

TUGAS AKHIR

KONVERSI SHARIF JUDGE DARI CODEIGNITER 3 KE CODEIGNITER 4



Filipus

NPM: 6181901074

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2023

DAFTAR ISI

DAFTAR ISI	iii
DAFTAR GAMBAR	vii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 CodeIgniter 3[1]	5
2.1.1 <i>Model-View-Controller</i>	5
2.1.2 <i>CodeIgniter URLs</i>	7
2.1.3 <i>Helpers</i>	8
2.1.4 <i>Libraries</i>	9
2.1.5 <i>Database</i>	14
2.1.6 <i>URI Routing</i>	15
2.1.7 <i>Auto-loading</i>	16
2.2 SharIF Judge[2]	16
2.2.1 Struktur Aplikasi	16
2.2.2 Instalasi	16
2.2.3 <i>Clean URLs</i>	18
2.2.4 Users	18
2.2.5 Menambah <i>Assignment</i>	19
2.2.6 <i>Sample Assignment</i>	23
2.2.7 <i>Test Structure</i>	25
2.2.8 Deteksi Kecurangan	27
2.2.9 Keamanan	28
2.2.10 <i>Sandboxing</i>	30
2.2.11 <i>Shield</i>	30
2.3 CodeIgniter 4[3]	32
2.3.1 <i>Models-Views-Controllers</i>	33
2.3.2 <i>Autoloading Files</i>	36
2.3.3 <i>Configuration</i>	37
2.3.4 <i>CodeIgniter URLs</i>	38
2.3.5 <i>URI Routing</i>	39
2.3.6 <i>Database</i>	39
2.3.7 <i>Library</i>	41
2.3.8 <i>Helpers</i>	46
2.4 Konversi CodeIgniter 3 ke CodeIgniter 4[3]	46

2.4.1	Struktur Aplikasi	46
2.4.2	<i>Routing</i>	47
2.4.3	<i>Model, View, and Controller</i>	47
2.4.4	<i>Class Loading</i>	48
2.4.5	<i>Configuration</i>	48
2.4.6	<i>Database</i>	49
2.4.7	<i>Migrations</i>	49
2.4.8	<i>Routing</i>	49
2.4.9	<i>Libraries</i>	49
2.4.10	<i>Helpers</i>	51
2.4.11	<i>Events</i>	51
2.4.12	<i>Framework</i>	51
3	ANALISIS	53
3.1	Analisis Sistem Kini	53
3.1.1	<i>Model</i>	53
3.1.2	<i>View</i>	58
3.1.3	<i>Controller</i>	66
3.1.4	<i>Assets</i>	72
3.1.5	<i>Config</i>	73
3.1.6	<i>Libraries</i>	74
3.2	Analisis Sistem Usulan	78
3.2.1	Persiapan <i>CodeIgniter 4</i>	78
3.2.2	Struktur Aplikasi	78
3.2.3	<i>Routing</i>	80
3.2.4	<i>Model, View, and Controller</i>	80
3.2.5	<i>Libraries</i>	83
3.2.6	<i>Configuration</i>	87
3.2.7	<i>Database</i>	88
3.2.8	<i>Helpers</i>	88
4	PERANCANGAN	91
4.1	Instalasi <i>CodeIgniter 4</i>	91
4.2	Perubahan Struktur Aplikasi	91
4.2.1	<i>app/Config</i>	91
4.2.2	<i>Controllers</i>	98
4.2.3	<i>Filters</i>	100
4.2.4	<i>Helpers</i>	103
4.2.5	<i>Libraries</i>	103
4.2.6	<i>Models</i>	110
4.2.7	<i>View</i>	112
4.2.8	<i>public</i>	114
4.2.9	<i>restriced</i>	114
5	IMPLEMENTASI DAN PENGUJIAN	115
5.1	Lingkungan Implementasi dan Pengujian	115
5.2	Implementasi	116
5.2.1	Kode Program	116
5.2.2	Basis Data	116
5.3	Pengujian Fungsional	117
6	KESIMPULAN DAN SARAN	125

6.1	Kesimpulan	125
6.2	Saran	125
DAFTAR REFERENSI		127
A	KODE PROGRAM <i>Controllers</i>	129
B	KODE PROGRAM <i>Models</i>	171
C	KODE PROGRAM <i>Views</i>	195

DAFTAR GAMBAR

1.1	Tampilan halaman <i>SharIF Judge</i>	1
1.2	<i>Pemindahan struktur aplikasi menuju CodeIgniter 4</i>	2
2.1	<i>Flow Chart Aplikasi CodeIgniter 3</i>	5
2.2	Tampilan halaman <i>SharIF Judge</i> untuk menambahkan <i>assignment</i>	20
3.1	Tampilan Halaman <i>Dashboard</i>	58
3.2	Tampilan Halaman <i>Profile</i>	59
3.3	Tampilan Halaman <i>Settings</i>	59
3.4	Tampilan Halaman <i>Users</i>	60
3.5	Tampilan Halaman <i>Notifications</i>	60
3.6	Tampilan Halaman <i>Assignments</i>	61
3.7	Tampilan Halaman <i>Problems</i>	61
3.8	Tampilan Halaman <i>Submit</i>	62
3.9	Tampilan Halaman <i>Final Submission</i>	62
3.10	Tampilan Halaman <i>All Submission</i>	63
3.11	Tampilan Halaman <i>Scoreboard</i>	63
3.12	Tampilan Halaman <i>Hall of Fame</i>	64
3.13	Tampilan Halaman <i>24-hour Log</i>	64
3.14	Tampilan Halaman <i>ReJudge</i>	65
3.15	Tampilan Halaman <i>Submission Queue</i>	65
3.16	Tampilan Halaman <i>Cheat Detection</i>	66
3.17	<i>Pemindahan struktur aplikasi menuju CodeIgniter 4</i>	79

1

BAB 1

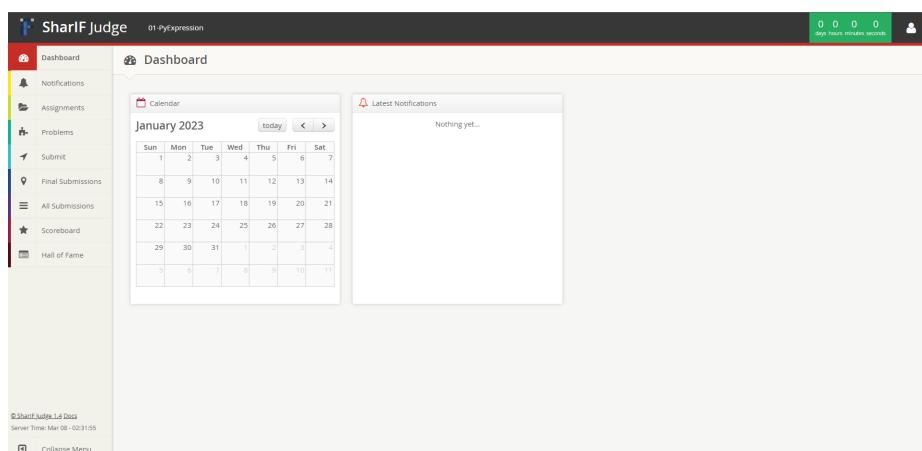
2

PENDAHULUAN

3 1.1 Latar Belakang

4 Tugas merupakan suatu bentuk pembelajaran dan penilaian yang diberikan oleh pengajar kepada
5 pelajar untuk membantu pelajar mendalami materi yang sudah diberikan[4]. Pembagian tugas
6 yang diberikan dapat dibagi menjadi 2 jenis yakni tugas individu dan tugas kelompok. Tugas
7 individu merupakan tugas yang hanya ditanggung oleh satu individu sedangkan, tugas kelompok
8 merupakan tugas yang ditanggung oleh beberapa individu. Tugas selanjutnya akan dikumpulkan
9 kepada pengajar dan diberikan penilaian berdasarkan tingkat ketepatan jawaban dari tugas tersebut.
10 Pengumpulan dan pengecekan tugas terutama *coding* secara manual memiliki kekurangan dimana
11 diperlukan banyak langkah dalam melakukan pengecekan dan pengiriman nilai. Pengecekan secara
12 manual juga terdapat kesulitan dalam pengecekan yakni, kekurangan dalam pengecekan plagiat
13 antara tugas pelajar. Maka, dibutuhkan perangkat lunak untuk melakukan pengecekan secara
14 otomatis salah satunya adalah *Online Judge*.

15 *Online Judge* merupakan sebuah perangkat lunak yang dapat melakukan pengecekan program
16 sesuai dengan standar yang sudah diberikan. Perangkat lunak ini dapat menerima jawaban dari
17 pelajar dan melakukan pengecekan secara otomatis dan memberikan keluaran berupa nilai dari
18 pelajar tersebut[5]. Salah satu perangkat lunak *Online Judge* terdapat pada Universitas Katolik
19 Parahyangan prodi Informatika bernama *SharIF Judge* (dapat dilihat pada Gambar 1.1).



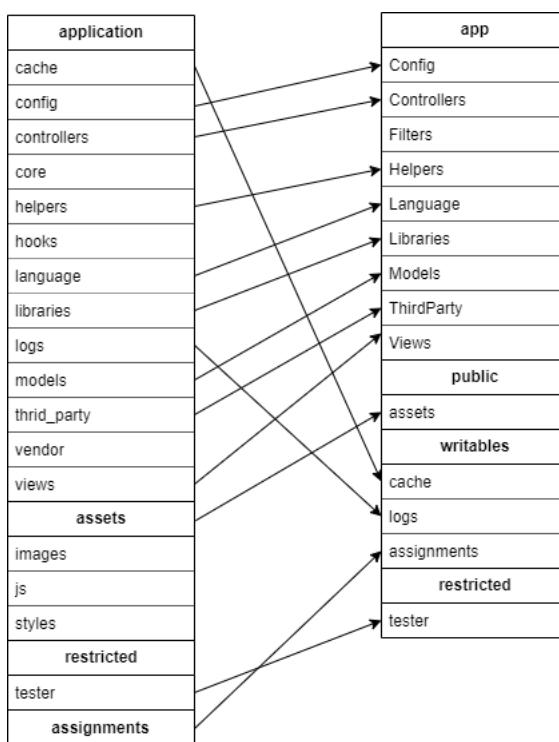
Gambar 1.1: Tampilan halaman *SharIF Judge*

20 *SharIF Judge* pada awalnya bernama *Sharif Judge* yang merupakan sebuah perangkat lunak *open*
21 *source* untuk menilai kode dengan beberapa bahasa seperti C, C++, Java, dan Python secara online.

1 Sharif Judge pada awalnya dibentuk oleh Mohammad Javad menggunakan *framework CodeIgniter 3*
 2 yang merupakan *framework* berbasis PHP (*Hypertext Preprocessor*). *Sharif Judge* kemudian di *fork*
 3 dan dimodifikasi menjadi *SharIF Judge* dengan penambahan fungsi sesuai kebutuhan Informatika
 4 Unpar untuk mengumpulkan tugas dan ujian mahasiswa[2].

5 *CodeIgniter 3* merupakan sebuah *framework opensource* yang bertujuan untuk mempermudah dalam membangun sebuah aplikasi *website* menggunakan PHP. *CodeIgniter 3* menggunakan struktur MVC yang membagi file menjadi 3 buah yaitu Model, View, Controller. Selain itu, *CodeIgniter 3* merupakan *framework* ringan dan menyediakan banyak *library* untuk digunakan oleh penggunanya[1]. Namun, *CodeIgniter 3* sudah memasuki fase *maintenance*¹ sehingga tidak mendapatkan pembaharuan lebih lanjut dari pembentuknya. *CodeIgniter 3* pada akhirnya tidak dapat dipakai dan akan kehilangan dokumentasi dari situs web resminya. Sehingga, perangkat lunak yang menggunakan *CodeIgniter 3* perlu dikonversi ke *framework* *CodeIgniter* dengan versi terbaru yakni *CodeIgniter 4*.

14 *CodeIgniter 4* merupakan versi terbaru dari *framework* *CodeIgniter* yang memiliki banyak perubahan fitur dari versi sebelumnya. *CodeIgniter 4* dapat dijalankan menggunakan versi PHP 7.4 atau lebih baru sedangkan *CodeIgniter 3* dapat dijalankan menggunakan versi PHP 5.6 atau lebih baru. *CodeIgniter 4* juga membagi file menggunakan struktur MVC namun, memiliki struktur direktori berbeda dengan versi sebelumnya[3]. Perubahan direktori dapat dilihat pada gambar 1.2.



Gambar 1.2: Pemindahan struktur aplikasi menuju *CodeIgniter 4*

19 Gambar 1.2 merupakan perubahan struktur yang terdapat pada *CodeIgniter 4*. Rincian perubahan dapat dilihat pada bab 3

21 Pada skripsi ini, akan dilakukan konversi *SharIF Judge* dari *CodeIgniter 3* sehingga dapat berjalan pada *CodeIgniter 4*.

¹Pemberitahuan fase *maintenance* *CodeIgniter 3* <https://codeigniter.com/download>(19 Maret 2023)

1 1.2 Rumusan Masalah

- 2 • Bagaimana cara melakukan konversi SharIF Judge pada CodeIgniter 3 menjadi CodeIgniter
3 4?
4 • Bagaimana mengevaluasi kode SharIF Judge pada CodeIgniter 3 dan mengubahnya agar dapat
5 berjalan pada CodeIgniter 4?

6 1.3 Tujuan

- 7 Tujuan dari skripsi ini adalah sebagai berikut:
- 8 • Melakukan konversi dengan megubah kode sesuai dengan standar CodeIgniter 4.
9 • Melakukan evaluasi kode SharIF Judge dan mengubahnya agar dapat berjalan di CodeIgniter
10 4.

11 1.4 Batasan Masalah

- 12 Batasan masalah pada pembentukan skripsi ini adalah sebagai berikut:
- 13 • Auntentikasi *RADIUS* tidak akan diimplementasi karena tidak memiliki *server* yang terinte-
14 grasi.

15 1.5 Metodologi

- 16 Metodologi yang dilakukan dalam melakukan penelitian ini adalah sebagian berikut:
- 17 1. Melakukan analisis dan eksplorasi fungsi-fungsi perangkat lunak SharIF Judge.
18 2. Melakukan studi literatur kebutuhan konversi dari CodeIgniter 3 menjadi CodeIgniter 4.
19 3. Melakukan konversi perangkat lunak dari CodeIgniter 3 menjadi CodeIgniter 4.
20 4. Melakukan pengujian dan eksperimen terhadap perangkat lunak yang sudah di konversi.
21 5. Menyelesaikan pembentukan dokumen

22 1.6 Sistematika Pembahasan

- 23 Penelitian ini akan dibahas dalam enam bab yang masing-masing berisi:

24 1. **Bab 1:** Pendahuluan

25 Bab ini berisi latar belakang, rumusan masalah,tujuan, batasan masalah, metodologi, dan
26 sistematika pembahasan.

27 2. **Bab 2:** Landasan Teori

28 Bab ini berisi pembahasan dasar-dasar teori yang akan digunakan dalam melakukan konversi
29 SharIF Judge dari CodeIgniter 3 ke CodeIgniter 4. Landasan Teori yang digunakan diantaranya
30 adalah SharIF Judge, CodeIgniter 3, CodeIgniter 4, dan Konversi CodeIgniter 3 ke CodeIgniter
31 4.

32 3. **Bab 3:** Analisis

33 Bab ini berisi analisis *SharIF Judge* dan analisis kebutuhan konversi menuju *CodeIgniter 3*.

1 **4. Bab 4:** Perancangan

2 Bab ini berisikan mengenai rancangan perangkat lunak yang akan dikonversi.

3 **5. Bab 5:** Implementasi dan Pengujian

4 Bab ini berisikan hasil implementasi dan pengujian yang telah dilakukan untuk melakukan
5 konversi *SharIF Judge* dari *CodeIgniter 3* ke *CodeIgniter 4*.

6 **6. Bab 6:** Kesimpulan dan Saran

7 Bab ini berisikan kesimpulan dari hasil konversi yang telah dilakukan dan saran-saran terhadap
8 perangkat lunak.

1

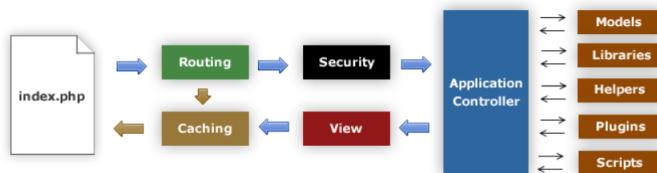
BAB 2

2

LANDASAN TEORI

3 2.1 CodeIgniter 3[1]

4 CodeIgniter 3 merupakan sebuah *framework opensource* yang berfungsi untuk mempermudah
5 pengguna dalam membangun aplikasi *website* menggunakan bahasa PHP. CodeIgniter 3 memiliki
6 tujuan untuk membantu pengguna dalam membangun aplikasi web lebih cepat dengan menyediakan
7 beragam *library* dan tampilan dan *logic* yang simpel. *CodeIgniter 3* juga merupakan *framework*
8 ringan yang menggunakan struktur *Model-View-Controller*, dan menghasilkan *URLs* yang bersih.
9 *Code Igniter 3* memiliki *flow chart* aplikasi yang dapat dilihat pada gambar 2.1.



Gambar 2.1: *Flow Chart* Aplikasi *CodeIgniter 3*

10 Berikut merupakan pembagian *flow chart* aplikasi *CodeIgniter 3*:

- 11 1. *index.php* berfungsi sebagai *front controller* yang berguna untuk melakukan inisiasi
- 12 2. *Router* berfungsi dalam melakukan pemeriksaan dan menentukan penggunaan *HTTP Request*.
- 13 3. *Cache* berfungsi untuk mengirimkan *file cache*(apabila ada) kepada *browser* secara langsung.
- 14 4. *Security* berfungsi sebagai alat penyaringan setiap data dan *HTTP Request* yang masuk. Penyaringan data tersebut dilakukan sebelum *controller* aplikasi dimuat agar aplikasi menjadi lebih aman.
- 15 5. *Controller* berguna sebagai alat untuk memuat *model*, *libraries*, dan sumber daya yang dibutuhkan untuk menjalankan permintaan spesifik.
- 16 6. *View* akan dikirimkan menuju *browser* untuk dilihat oleh pengguna. Apabila *caching* dinyalakan, maka *view* akan dilakukan *cached* terlebih dahulu sehingga permintaan selanjutnya dapat diberikan.

22 2.1.1 *Model-View-Controller*

23 *CodeIgniter 3* merupakan *framework* berbasis arsitektur *Model-View-Controller* atau yang selanjutnya akan disebut sebagai *MVC*. *MVC* merupakan sebuah pendekatan perangkat lunak yang

¹ memisahkan antara logika dengan presentasi atau tampilannya. Penggunaan struktur ini mengu-
² rangi penggunaan skrip pada halaman web karena tampilan terpisah dengan skrip PHP. Berikut
³ merupakan penjelasan mengenai struktur MVC:

4 *Model*

Model berfungsi dalam mewakili struktur data perangkat lunak. *Model* berfungsi dalam mengambil, memasukan, dan memperbarui data pada *database*. Berikut merupakan contoh *file Model CodeIgniter* pada direktori **application/models/**:

Kode 2.1: Contoh *model* pada *CodeIgniter 3*

```
8 class Blog_model extends CI_Model {
9
10
11     public $title;
12     public $content;
13     public $date;
14
15     public function get_last_ten_entries()
16     {
17         $query = $this->db->get('entries', 10);
18         return $query->result();
19     }
20
21     public function insert_entry()
22     {
23         $this->title    = $_POST['title']; // please read the below note
24         $this->content  = $_POST['content'];
25         $this->date     = time();
26
27         $this->db->insert('entries', $this);
28     }
29
30     public function update_entry()
31     {
32         $this->title    = $_POST['title'];
33         $this->content  = $_POST['content'];
34         $this->date     = time();
35
36         $this->db->update('entries', $this, array('id' => $_POST['id']));
37     }
38 }
```

⁴¹ Kode 2.1 terdapat beberapa fungsi yaitu:

- `get_last_ten_entries()` yang berfungsi untuk mengambil 10 data terakhir dari tabel `entries` menggunakan *query builder*.
 - `insert_entry()` yang berfungsi untuk memasukan data *title*,*content*,dan *date* menuju tabel `entries`.
 - `update_entry()` yang befungsi untuk memperbarui data *title*,*content*,dan *date* pada tabel `entries`.

⁴⁸ Model biasanya digunakan pada *file controller* dan dapat dipanggil menggunakan kode sebagai berikut:

```
50         $this->load->model('model name');
```

51 *View*

52 *View* berfungsi dalam menyajikan informasi kepada pengguna. *View* biasanya merupakan halaman
53 web namun, pada *CodeIgniter 3 view* dapat berupa pecahan halaman seperti *header* atau *footer*.
54 Pecahan halaman dapat dimasukan pada halaman lain agar mempermudah dan membangun kode
55 yang lebih bersih.

Kode 2.2: Contoh *view* pada *CodeIgniter 3*

```

21 <?php
32 <html>
43 <head>
54     <title>My Blog</title>
65 </head>
76 <body>
87     <h1>Welcome to my Blog!</h1>
98 </body>
109 </html>

```

12 Kode 2.2 merupakan contoh *file view* *CodeIgniter 3* pada direktori **application/views/** yang
 13 berisikan judul My Blog dan *heading* Welcome to my Blog!. Pengguna dapat memanggil halaman
 14 yang sudah dibentuk pada *file controller* dengan cara sebagai berikut:

```
15 $this->load->view('name');
```

16 *Controller*

17 *Controller* berfungsi sebagai perantara antara *Model*, *View*, dan sumber daya yang dibutuhkan untuk
 18 melakukan proses *HTTP Request* dan menjalankan halaman web. Penamaan *controller* biasanya
 19 digunakan sebagai *url* pada perangkat lunak pengguna. Berikut merupakan contoh *controller*
 20 *CodeIgniter 3* pada direktori **application/controllers/**:

Kode 2.3: Contoh *controller* pada *CodeIgniter 3*

```

21 <?php
22 class Blog extends CI_Controller {
23
24     public function index()
25     {
26         echo 'Hello World!';
27     }
28
29     public function comments()
30     {
31         echo 'Look at this!';
32     }
33 }
34

```

36 Kode 2.3 berfungsi dalam mengembalikan *string* sesuai dengan fungsi *controller* yang dipanggil
 37 .Nama *controller* dan metode diatas akan dijadikan segmen pada *URL* seperti berikut:

```
38 example.com/index.php/blog/index/
```

39 Metode *index* akan secara otomatis dipanggil menjadi *URL* dan pengguna juga dapat memberi
 40 parameter untuk metode *controller* yang nantinya dapat diambil dari *URL*.

41 **2.1.2 CodeIgniter URLs**

42 *CodeIgniter 3* menggunakan pendekatan *segment-based* dibandingkan menggunakan *query string*
 43 untuk membangun *URL* yang mempermudah mesin pencari dan pengguna. Berikut merupakan
 44 contoh *URL* pada *CodeIgniter 3*:

```
45 example.com/news/article/my_article
```

46 *URL* diatas dibentuk berdasarkan *Controller* sebagai berikut :

```
47 example.com/class/function/ID
```

- 1 Segmen tersebut dibagi menjadi tiga buah yakni:
- 2 1. Segmen pertama merepresentasikan kelas *controller* yang dipanggil.
- 3 2. Segmen kedua merepresentasikan kelas fungsi atau metode yang digunakan.
- 4 3. Segmen ketiga dan segmen lainnya merepresentasikan *ID* dari variabel yang akan dipindahkan
5 menuju *controller*.

6 Secara asali *URL* yang dihasilkan *CodeIgniter 3* terdapat nama *file index.php* seperti contoh
7 dibawah ini:

```
8 example.com/index.php/news/article/my_article
```

9 Pengguna dapat menghapus *file index.php* file pada *url* menggunakan *file .htaccess* apabila
10 *server Apache* pengguna menghidupkan *mod_rewrite*. Berikut merupakan contoh *file .htaccess*
11 menggunakan metode *negative*:

Kode 2.4: Contoh *file .htaccess* pada halaman index.php

```
12
13 1 RewriteEngine On
14 2 RewriteCond %{REQUEST_FILENAME} !-f
15 3 RewriteCond %{REQUEST_FILENAME} !-d
16 4 RewriteRule ^(.*)$ index.php/$1 [L]
```

18 Aturan diatas menyebabkan *HTTP Request* selain yang berasal dari direktori atau *file* diperlakukan
19 sebagai sebuah permintaan pada file *index.php*. Selain itu, pengguna juga dapat menambahkan
20 akhiran pada *URL* agar halaman pengguna dapat menampilkan halaman sesuai dengan tipe yang
21 diingikan. Berikut merupakan contoh *URL* sebelum dan sesudah ditambahkan akhiran berupa:

```
22 example.com/index.php/products/view/shoes
23 example.com/index.php/products/view/shoes.html
```

24 Pengguna juga dapat menyalakan fitur *query strings* dengan cara sebagai mengubah *file application/config.php* seperti berikut:

Kode 2.5: *File application/config.php*

```
26
27 1 $config['enable_query_strings'] = FALSE;
28 2 $config['controller_trigger'] = 'c';
29 3 $config['function_trigger'] = 'm';
```

31 Pengguna dapat mengubah *enable_query_strings* menjadi *TRUE*. Pengubahan fitur tersebut
32 dapat memperbolehkan pengguna untuk menambahkan *query* pada *URL* yang dibentuk. Berikut
33 merupakan contoh pengaksesan *URL* melalui *query strings*:

```
34 index.php?c=controller&m=method
```

35 *URL* dapat mengakses fungsi ataupun metode dari *controller* menggunakan *query* tanpa harus
36 menggunakan *URL* biasa.

37 2.1.3 *Helpers*

38 *Helpers* merupakan fungsi pada *CodeIgniter 3* yang mempermudah pengguna dalam membangun
39 aplikasi web. Setiap *file helpers* terdiri dari banyak fungsi yang membantu sesuai kategori dan tidak
40 ditulis dalam format *Object Oriented*. *File helpers* terdapat pada direktori *system/helpers* atau
41 *application/helpers*. Pengguna dapat memakai fitur *helpers* dengan cara memuatnya seperti berikut:

```
1     $this->load->helper('name');
```

2 Pemanggilan *helper* tidak menggunakan ekteksi .php melainkan hanya menggunakan nama dari
 3 *helper* yang ingin digunakan. Pengguna dapat memanggil satu atau banyak *helper* pada metode
 4 *controller* ataupun *view* sesudah dimuat.

5 2.1.4 *Libraries*

6 *CodeIgniter 3* menyediakan *library* yang dapat dipakai pengguna untuk mempermudah pembentukan
 7 aplikasi web. *Library* merupakan kelas yang tersedia pada direktori *application/libraries* dan dapat
 8 ditambahkan, diperluas, dan digantikan.

Kode 2.6: Contoh kelas *library* pada *CodeIgniter 3*

```
9 <?php
10 defined('BASEPATH') OR exit('No direct script access allowed');
11
12 class Someclass {
13
14     public function some_method()
15     {
16
17     }
18
19 }
```

20 Kode 2.6 merupakan contoh *file library* pada *CodeIgniter 3*. Setiap pembentukan *file library*
 21 diperlukan huruf kapital dan harus sama dengan nama klasnya. Berikut merupakan contoh
 22 pemanggilan *file library* pada *file controller*:

```
23     $params = array('type' => 'large', 'color' => 'red');
24     $this->load->library('someclass', $params);
```

25 Kode diatas memanggil *library someclass* dan dapat dilakukan melalui *controller* manapun
 26 dan dapat diberikan parameter sesuai dengan metode yang dibentuk pada *library*. *CodeIgniter 3*
 27 menyediakan berbagai *library* yang dapat digunakan oleh pengguna seperti berikut.

28 *Kelas Input*

29 Kelas *input* memiliki dua buah fungsi yakni melakukan praproses data masukan dan memberikan
 30 metode kepada beberapa *helper*. Penggunaan kelas *input* dapat dipanggil menggunakan sintaks
 31 berikut.

```
32     $something = $this->input->post('something');
```

33 Sintaks diatas akan mengambil data yang dikirim menggunakan *post* pada data bernama *something*.

34 *JavaScript*

35 Penggunaan kelas *javascript* dapat dipanggil pada konstruktor *controller* dengan cara berikut.

```
36     $this->load->library('javascript');
```

37 Sintaks diatas akan melakukan inisiasi *library javascript* pada file tersebut. Pengguna selanjutnya
 38 harus melakukan inisiasi *library* pada halaman *view tag <head>* seperti berikut.

```

1 <?php echo $library_src;?>
2 <?php echo $script_head;?>
```

3 Sintaks `$library_src` merupakan lokasi *library* yang akan dimuat sedangkan `$script_head`
4 merupakan lokasi untuk fungsi yang akan dijalankan. Selanjutnya *javascript* dapat diinisiasi
5 pada *controller* menggunakan sintaks berikut:

```
6 $this->javascript
```

7 Selain menggunakan *javascript*, pengguna dapat memakai *jQuery* dengan menambahkan *jQuery*
8 pada akhir inisiasi kelas *javascript* seperti berikut:

```
9 $this->load->library('javascript/jquery');
```

10 Pengguna dapat memakai berbagai fungsi *library jquery* menggunakan sintaks berikut:

```
11 $this->jquery
```

12 Kelas *Email*

13 *CodeIgniter 3* menyediakan kelas *email* dengan fitur sebagai berikut:

- 14 • Beberapa Protokol: *Mail*, *Sendmail*, dan *SMTP*
- 15 • Enkripsi *TLS* dan *SSL* untuk *SMTP*
- 16 • Beberapa Penerima
- 17 • *CC* dan *BCCs*
- 18 • *HTML* atau *email* teks biasa
- 19 • Lampiran
- 20 • Pembungkus kata
- 21 • Prioritas
- 22 • Mode *BCC Batch*, memisahkan daftar *email* besar menjadi beberapa *BCC* kecil.
- 23 • Alat *Debugging email*

24 Penggunaan *library email* dapat dikonfigurasikan pada *file config*. Pengguna dapat mengirim
25 *email* dengan mudah menggunakan fungsi-fungsi yang telah disediakan *library email*. Kode 2.7
26 merupakan contoh pengiriman *email* melalui *controller*.

Kode 2.7: Contoh pengiriman *email* melalui *controller*

```

27 $this->load->library('email');
28
29
30 $this->email->from('your@example.com', 'Your Name');
31 $this->email->to('someone@example.com');
32 $this->email->cc('another@another-example.com');
33 $this->email->bcc('them@their-example.com');
34
35 $this->email->subject('Email Test');
36 $this->email->message('Testing the email class.');
37
38 $this->email->send();
```

40 Kode 2.7 merupakan contoh penggunaan *library email* untuk mengirim *email* dari `your@example.com`
41 menuju `someone@example.com`. Konfigurasi ini juga mengirim dua buah salinan menuju `another@`
42 `another-example.com` dan `them@their-example.com` dengan subjek berupa *Email Test* dengan

¹ pesan `Testing the email class.` Selain itu, pengguna juga dapat melakukan konfigurasi preferensi *email* melalui dua puluh satu preferensi. Pengguna dapat melakukan konfigurasi secara otomatis melalui *file config* atau melakukan konfigurasi secara manual . Kode [2.8](#) merupakan contoh ⁴ konfigurasi preferensi secara manual.

Kode 2.8: Contoh konfigurasi preferensi *library email* secara manual

```
5 $config['protocol'] = 'sendmail';
6 $config['mailpath'] = '/usr/sbin/sendmail';
7 $config['charset'] = 'iso-8859-1';
8 $config['wordwrap'] = TRUE;
9
10
11 $this->email->initialize($config);
12
```

¹³ Kode 2.8 merupakan contoh konfigurasi pengiriman *email* dengan protokol `sendmail`, tujuan
¹⁴ `usr/sbin/sendmail`, *character set* berjenis `iso-8859-1`, dan menyalakan fitur `wordwrap`. Selan-
¹⁵ jutnya konfigurasi dapat diinisialisasikan menggunakan `initialize`.

16 Kelas *File Uploading*

¹⁷ Pengunggahan file terdapat empat buah proses sebagai berikut:

- 18 1. Dibentuk sebuah form untuk pengguna memilih dan mengunggah *file*.
19 2. Setelah *file* diunggah, *file* akan dipindahkan menuju direktori yang dipilih.
20 3. Pada pengiriman dan pemindahan *file* dilakukan validasi sesuai dengan ketentuan yang ada.
21 4. Setelah *file* diterima akan dikeluarkan pesan berhasil.

22 Perangkat lunak akan memindahkan *file* yang sudah diunggah pada *form* menuju *controller*
23 untuk dilakukan validasi dan penyimpanan.

Kode 2.9: Contoh *controller* untuk melakukan validasi dan penyimpanan

```
24
25 1 <?php
26 2
27 3 class Upload extends CI_Controller {
28 4
29 5     public function __construct()
30 6     {
31 7         parent::__construct();
32 8         $this->load->helper(array('form', 'url'));
33 9     }
34 0
35 1     public function index()
36 2     {
37 3         $this->load->view('upload_form', array('error' => ' '));
38 4     }
39 5
40 6     public function do_upload()
41 7     {
42 8         $config['upload_path']          = './uploads/';
43 9         $config['allowed_types']        = 'gif|jpg|png';
44 0         $config['max_size']             = 100;
45 1         $config['max_width']            = 1024;
46 2         $config['max_height']           = 768;
47 3
48 4         $this->load->library('upload', $config);
49 5
50 6         if ( ! $this->upload->do_upload('userfile'))
51 7         {
52 8             $error = array('error' => $this->upload->display_errors());
53 9
54 0             $this->load->view('upload_form', $error);
55 1         }
56 2     else
57 3     {
58 4         $data = array('upload_data' => $this->upload->data());
59 5
60 6             $this->load->view('upload_success', $data);
61 7     }
62 8 }
```

```

28 }
29 }
30 ?>

```

5 Kode 2.9 merupakan contoh kode dengan dua buah metode yakni:

- 6 1. `index()` yang digunakan untuk mengembalikan *view* bernama `upload_form`
- 7 2. `do_upload` yang digunakan untuk melakukan validasi berupa tipe, ukuran, lebar, dan panjang
- 8 maksimal sebuah file. Metode ini juga mengembalikan *error* dan menyimpan *file* pada direktori
- 9 `./uploads/`.

10 Direktori penyimpanan dapat diubah sesuai dengan kebutuhan namun perlu pengubahan izin
11 direktori menjadi 777 agar dapat di baca, di tulis, dan di eksekusi oleh seluruh pengguna.

12 Session

13 *Library session* memperbolehkan aplikasi untuk melihat jejak dari aktivitas pengguna. *CodeIgniter*
14 menyediakan beberapa cara penyimpanan *session* yakni menggunakan file, *database*, *redis*, dan
15 *memcached*. *Library* ini dapat diinisiasi menggunakan sintaks berikut:

```

16         $this->load->library('session');

```

17 Sintaks diatas akan melakukan inisiasi terhadap *library session* dan dapat digunakan menggunakan
18 sintaks berikut:

```

19         $this->session

```

20 *Session* bekerja ketika halaman termuat dan akan melakukan pengecekan terhadap *session cookie*
21 pengguna. Data ini akan disimpan dan diperbarui sesuai dengan data yang diterima. Pengguna
22 dapat mengambil data dari *session* ini menggunakan sintaks sebagai berikut:

```

23         $this->session->userdata('item');

```

24 Sintaks diatas akan mengambil seluruh data *session* dengan nama *item*. Secara *default*, aplikasi
25 akan menyimpan data *session* pada file namun pengguna dapat mengubah konfigurasi sesuai dengan
26 kebutuhan. Berikut merupakan konfigurasi untuk menyimpan data pada *database*.

Kode 2.10: Contoh konfiguras untuk menyimpan data pada *database*

```

27 1 $config['sess_driver'] = 'database';
28 2 $config['sess_save_path'] = 'ci_sessions';

```

31 Kode 2.10 merupakan konfigurasi untuk menyimpan data menuju *database* dengan nama tabel
32 *ci_sessions*.

33 Kelas URI

34 Kelas *URI* menyediakan metode untuk mengambil informasi yang terdapat pada *URI strings*.
35 Kelas ini diinisiasi secara otomatis oleh *CodeIgniter* sehingga tidak perlu dilakukan inisiasi lagi.
36 Informasi dapat diambil menggunakan sintaks berikut:

```

37         $array = $this->uri->uri_to_assoc(3, $default);

```

38 Sintaks diatas akan mengambil informasi yang tersedia pada *URI segment* menjadi sebuah *associative*
39 *array*. *Array* yang dihasilkan akan berisikan *key* dan *value* sesuai dengan penempatan *segment*.

1 Kelas *Zip Encoding*

2 *Zip* merupakan format sebuah *file* yang berguna untuk melakukan kompress terhadap *file* untuk
 3 mengurangi ukuran dan menjadikannya sebuah *file*. *CodeIgniter 3* menyediakan *library Zip Encoding*
 4 yang dapat digunakan untuk membangun arsip *Zip* yang dapat diunduh menuju *desktop* atau
 5 disimpan pada direktori. *Library* ini dapat diinisiasi dengan kode sebagai berikut:

```
6           $this->load->library('zip');
```

7 Setelah diinisiasi, pengguna dapat memanggil *library* tersebut menggunakan kode sebagai
 8 berikut:

```
9           $this->zip
```

10 Kode 2.11 merupakan contoh penggunaan *library Zip Encoding* untuk menyimpan dan menunduh
 11 data.

Kode 2.11: Contoh penggunaan *library Zip Encoding*

```
12 $name = 'mydata1.txt';
13 $data = 'A Data String!';
14
15
16 $this->zip->add_data($name, $data);
17
18 // Write the zip file to a folder on your server. Name it "my_backup.zip"
19 $this->zip->archive('/path/to/directory/my_backup.zip');
20
21 // Download the file to your desktop. Name it "my_backup.zip"
22 $this->zip->download('my_backup.zip');
```

24 Kode 2.11 merupakan contoh untuk melakukan penyimpanan *Zip file* pada direktori dan dapat
 25 mengunduh *file* menuju *desktop* pengguna. Selain menggunakan *library* yang sudah disediakan,
 26 pengguna dapat membangun dan memperluas *libraries* sendiri sesuai dengan kebutuhan. Kode
 27 merupakan contoh *library* yang dibentuk.

Kode 2.12: Contoh *library* yang dibentuk

```
28 class Example_library {
29
30     protected $CI;
31
32
33     // We'll use a constructor, as you can't directly call a function
34     // from a property definition.
35     public function __construct()
36     {
37         // Assign the CodeIgniter super-object
38         $this->CI =& get_instance();
39     }
40
41     public function foo()
42     {
43         $this->CI->load->helper('url');
44         redirect();
45     }
46 }
```

48 *Library* diatas merupakan contoh *library* yang dibentuk oleh pengguna digunakan untuk me-
 49 manggil *helper* bernama *url*. Pengguna dapat memanggil kelas tersebut seperti memanggil kelas
 50 *library* lainnya. Selain itu, pengguna juga dapat mengganti *library* yang sudah ada dengan *library*
 51 yang dibentuk pengguna dengan mengubah nama kelas sama persis dengan nama *library* yang ingin
 52 digantikan.

1 2.1.5 Database

² CodeIgniter 3 memiliki konfigurasi *database* yang menyimpan data-data terkait aturan *database*.

Kode 2.13: Contoh konfigurasi *database*

```

4 1 $db['default'] = array(
5 2     'dsn'      => '',
6 3     'hostname' => 'localhost',
7 4     'username' => 'root',
8 5     'password' => '',
9 6     'database'  => 'database_name',
10 7    'dbdriver'   => 'mysqli',
11 8    'dbprefix'  => '',
12 9    'pconnect'  => TRUE,
13 0    'db_debug'  => TRUE,
14 1    'cache_on'  => FALSE,
15 2    'cachedir'  => '',
16 3    'char_set'  => 'utf8',
17 4    'dbcollat'  => 'utf8_general_ci',
18 5    'swap_pre'  => '',
19 6    'encrypt'   => FALSE,
20 7    'compress'  => FALSE,
21 8    'stricton'  => FALSE,
22 9    'failover'  => array())
23 0 );
24 1

```

Kode 2.13 merupakan contoh konfigurasi pada file *database* untuk *database* bernama *database_name* dengan *username* *root* tanpa sebuah *password*. *CodeIgniter 3* menyediakan fitur *query* untuk menyimpan, memasukan, memperbarui, dan menghapus data pada *database* sesuai dengan konfigurasi *database* yang sudah diatur. Kode 2.14 merupakan contoh *query* untuk melakukan *select* dan *join* pada *CodeIgniter 3*:

Kode 2.14: Contoh penggunaan *query*

```
30
31 1 $this->db->select('*');
32 2 $this->db->from('blogs');
33 3 $this->db->join('comments', 'comments.id = blogs.id');
34 4 $query = $this->db->get();
```

36 Kode 2.14 merupakan contoh kode untuk melakukan *query* pada tabel **blogs** yang melakukan
37 *join* dengan tabel **comments**. Pengguna dapat mengambil hasil dari *query* menjadi *object* atau *array*.
38 Selain itu, *database* pada *CodeIgniter 3* juga dapat digunakan untuk membangun, menghapus,
39 dan mengubah *database* ataupun menambahkan kolom pada *table*. Penggunaan *database* untuk
40 membentuk, menghapus, atau mengubah *database* harus dilakukan inisiasi sebagai berikut:

```
41         $this->load->dbforge()
```

⁴² Setelah dilakukan inisiasi pengguna dapat membangun *database* menggunakan kelas *Forge*. Kode
⁴³ [2.15](#) merupakan contoh untuk membangun *database* dengan nama `db name`.

Kode 2.15: Contoh membangun *database* menggunakan *CodeIgniter3*

```
44 $this->dbforge->create_database('db_name')  
45 ]
```

⁴⁷ Selain itu, pengguna juga dapat menambahkan kolom dengan konfigurasinya. Kode 2.16
⁴⁸ merupakan contoh penambahan kolom sesuai dengan kebutuhan pengguna.

Kode 2.16: Contoh menambahkan kolom dengan konfigurasinya menggunakan *CodeJaniter3*

```
49 $fields = array(
50 1     'blog_id' => array(
51 2         'type' => 'INT',
52 3         'constraint' => 5,
53 4         'unsigned' => TRUE,
```

```

16         'auto_increment' => TRUE
27     ),
38     'blog_title' => array(
49         'type' => 'VARCHAR',
50         'constraint' => '100',
51         'unique' => TRUE,
62     ),
83     'blog_author' => array(
94         'type' => 'VARCHAR',
105         'constraint' => '100',
116         'default' => 'King of Town',
127     ),
138     'blog_description' => array(
149         'type' => 'TEXT',
120         'null' => TRUE,
161     ),
172 );
183 $this->dbforge->add_field($fields)
184 $this->dbforge->create_table('table_name');
```

21 Kode 2.16 merupakan contoh penggunaan *database* untuk menambahkan kolom sesuai dengan tipe,
 22 batas dari data yang disimpan, penambahan otomatis, dan lainnya pada tabel `table_name`.

23 2.1.6 *URI Routing*

24 *URL string* yang dihasilkan *CodeIgniter 3* biasanya menggunakan nama atau metode *controller*
 25 seperti pada berikut:

```
26           example.com/class/function/id/
```

27 Namun, pengguna dapat melakukan pemetaan ulang terhadap *url* yang dibentuk agar dapat
 28 memanggil metode dengan penambahan *segment* yang diinginkan.

Kode 2.17: Contoh *url* yang sudah dimetakan

```

29 1 example.com/product/1/
30 2 example.com/product/2/
31 3 example.com/product/3/
32 4 example.com/product/4/
```

35 Kode 2.17 merupakan contoh *url* yang sudah dimetakan ulang. Pengguna dapat menambahkan
 36 kode pemetaan pada file *application/config/routes.php* yang terdapat array bernama `$route`. Berikut
 37 merupakan beberapa cara melakukan pemetaan terhadap *url*:

38 WildCards

39 *Route wildcard* biasanya berisikan kode seperti berikut:

```
40             $route['product/:num'] = 'catalog/product_lookup';
```

41 *Route* diatas dibagi menjadi dua buah yakni:

42 1. Bagian segmen *URL*

43 Bagian pertama merupakan segmen pertama *url* yang akan tampil pada *url*. Bagian kedua
 44 merupakan segmen kedua dapat berisikan angka atau karakter.

45 2. Bagian kelas dan metode

46 Bagian kedua berisikan kelas dan metode dari *controller* yang akan digunakan pada *url*.

1 Ekspresi Reguler

2 Pengguna dapat memakai ekspresi reguler untuk melakukan pemetaan ulang *route*. Berikut
 3 merupakan contoh ekspresi reguler yang biasa digunakan:

4 \$route['products/([a-z]+)/(\d+)'] = '\$1/id_\$2';

5 Ekspresi ini menghasilkan *URI products/shirts/123* yang memanggil kelas *controller* dan metode
 6 *id_123*. Pengguna juga dapat mengambil segmen banyak seperti berikut:

7 \$route['login/(.+)'] = 'auth/login/\$1';

8 2.1.7 Auto-loading

9 *CodeIgniter 3* menyediakan sebuah fungsi untuk memuat berbagai kelas seperti *libraries*, *helpers*, dan
 10 *models*. Kelas dapat dimasukan pada *application/config/autoload.php* sesuai dengan petunjuk
 11 yang ada. *File autoload* akan diinisiasi setiap aplikasi dijalankan sehingga pengguna tidak perlu
 12 memuat kelas tersebut berulang kali.

13 2.2 SharIF Judge[2]

14 *SharIF Judge* merupakan sebuah *Online Judge* percabangan dari *Sharif Judge* yang dibentuk oleh
 15 Mohammed Javad Naderi. *Sharif Judge* dibentuk menggunakan *CodeIgniter 3* dan dimodifikasi
 16 sesuai dengan kebutuhan di Informatika Universitas Katolik Parahyangan menjadi nama *SharIF*
 17 *Judge*. *SharIF Judge* dapat menilai kode berbahasa *C*, *C++*, *Java*, dan *Python* dengan mengunggah
 18 file ataupun mengetiknya secara langsung.

19 2.2.1 Struktur Aplikasi

```
20
21 .
22 |--- application
23 |   |--- cache
24 |   |--- config
25 |   |--- controllers
26 |   |--- core
27 |   |--- helpers
28 |   |--- hooks
29 |   |--- language
30 |   |--- libraries
31 |   |--- logs
32 |   |--- models
33 |   |--- third_party
34 |   |--- vendor
35 |   |--- views
36 |--- assets
37 |   |--- images
38 |   |--- js
39 |   |--- styles
40 |--- restricted
41 |   |--- tester
42 |--- system
43 |--- assignments
```

45 2.2.2 Instalasi

46 Berikut merupakan persyaratan dan langkah-langkah melakukan *instalasi SharIF Judge*:

1 Persyaratan

- 2 SharIF Judge dapat dijalankan pada sistem operasi *Linux* dengan syarat sebagai berikut:

3 • Diperlukan *webserver* dengan versi PHP 5.3 atau lebih baru.

4 • Pengguna dapat menjalankan PHP pada *command line*. Pada *Ubuntu* diperlukan instalasi

5 paket *php5-cli*.

6 • *MySQL database* dengan ekstensi *Mysqli* untuk PHP atau *PostgreSQL database*.

7 • PHP harus memiliki akses untuk menjalankan perintah melalui fungsi *shell_exec*.

Kode 2.18: Kode untuk melakukah pengetesan fungsi

```
1 echo shell_exec("php_-v");
```

- *Tools* untuk melakukan kompilasi dan menjalankan kode yang dikumpulkan (*gcc*, *g++*, *javac*, *java*, *python2*, *python3*).
 - *Perl* disarankan untuk diinstalasi untuk alasan ketepatan waktu, batas memori, dan memaksimalkan batas ukuran pada hasil kode yang dikirim.

15 Instalasi

- 16 1. Mengunduh versi terakhir dari *SharIF Judge* dan melakukan *unpack* pada direktori *public*
17 *html*.
18 2. Memindahkan *folder system* dan *application* diluar direktori *public* dan mengubah *path* pada
19 *index.php*(Opsional).

Kode 2.19: Contoh *path* pada halaman index.php

```
1 $system_path = '/home/mohammad/secret/system';
2 application_folder = '/home/mohammad/secret/application';
```

- 24 3. Membangun *database MySql* atau *PostgreSql* untuk *SharIF Judge*. Jangan melakukan instalasi
25 paket koneksi *database* apapun untuk *C*, *C++*, *Java*, atau *Python*.

26 4. Mengatur koneksi *database* pada *file application/config/database.example.php* dan me-
27 nyimpannya dengan nama *database.php*. Pengguna dapat menggunakan awalan untuk nama
28 tabel.

Kode 2.20: Contoh pengaturan koneksi untuk *database*

```
1 /* Enter database connection settings here: */
2 'dbdriver' => 'postgre', // database driver (mysqli, postgre)
3 'hostname' => 'localhost', // database host
4 'username' => ' ', // database username
5 'password' => ' ', // database password
6 'database' => ' ', // database name
7 'dbprefix' => 'shi_', // table prefix
```

- 38 5. Mengatur *RADIUS server* dan *mail server* pada *file application/config/secrets.example.php*
39 dan menyimpannya dengan nama *secrets.php*.
40 6. Mengatur *application/cache/Twig* agar dapat ditulis oleh PHP.
41 7. Membuka halaman utama SharIF Judge pada *web browser* dan mengikuti proses instalasi.
42 8. Melakukan *Log in* dengan akun admin.
43 9. Memindahkan direktori *tester* dan *assignments* diluar direktori publik dan mengatur kedua
44 direktori agar dapat ditulis oleh PHP. Selanjutnya Menyimpan *path* kedua direktori pada

halaman *Settings*. Direktori `assignments` digunakan untuk menyimpan *file-file* yang diunggah agar tidak dapat diakses publik.

2.2.3 Clean URLs

Secara asali, `index.php` merupakan bagian dari seluruh *urls* pada SharIF judge. Berikut merupakan contoh dari *urls* SharIF Judge.

```
http://example.mjnaderi.ir/index.php/dashboard
http://example.mjnaderi.ir/index.php/users/add
```

Pengguna dapat menghapus `index.php` pada *url* dan mendapatkan *url* yang baik apabila sistem pengguna mendukung *URL rewriting*.

```
http://example.mjnaderi.ir/dashboard
http://example.mjnaderi.ir/users/add
```

Cara Mengaktifkan Clean URLs

- Mengganti nama *file .htaccess2* pada direktori utama menjadi *.htaccess*.
- Mengganti `$config['index_page'] = 'index.php';` menjadi `$config['index_page'] = '';` pada *file application\config\config.php*.

2.2.4 Users

Pada perangkat lunak SharIF Judge, pengguna dibagi menjadi 4 buah. Keempat pengguna tersebut adalah *Admins*, *Head Instructors*, *Instructors*, dan *Students*. Tabel 2.1 merupakan pembagian tingkat setiap pengguna.

Tabel 2.1: *Tabel tingkat pengguna*

<i>User Role</i>	<i>User Level</i>
<i>Admin</i>	3
<i>Head Instructor</i>	2
<i>Instructor</i>	1
<i>Student</i>	0

Setiap pengguna memiliki akses untuk aksi yang berbeda berdasarkan tingkatnya. Tabel 2.2 merupakan aksi yang dapat dilakukan oleh setiap pengguna.

Tabel 2.2: Tabel izin aksi setiap pengguna

Aksi	Admin	Head Instructor	Instructor	Student
Mengubah <i>Settings</i>	✓	✗	✗	✗
Menambah/Menghapus Pengguna	✓	✗	✗	✗
Mengubah Peran Pengguna	✓	✗	✗	✗
Menambah/Menghapus/Mengubah <i>Assignment</i>	✓	✓	✗	✗
Mengunduh <i>Test</i>	✓	✓	✗	✗
Menambah/Menghapus/Mengubah Notifikasi <i>Rejudge</i>	✓	✓	✗	✗
Melihat/ <i>Pause</i> /Melanjutkan/ <i>Submission Queue</i>	✓	✓	✗	✗
Mendeteksi Kode yang Mirip	✓	✓	✗	✗
Melihat Semua Kode	✓	✓	✓	✗
Mengunduh Kode Final	✓	✓	✓	✗
Memilih <i>Assignment</i>	✓	✓	✓	✓
<i>Submit</i>	✓	✓	✓	✓

1 Menambahkan Pengguna

- 2 Admin dapat menambahkan pengguna melalui bagian *Add User* pada halaman *Users*. Admin harus mengisi setiap informasi dimana baris yang diawali # merupakan komen dan setiap baris lainnya mewakili pengguna dengan sintaks berikut:

Kode 2.21: Contoh sintaks untuk menambahkan pengguna

```

5 1 USERNAME,EMAIL,DISPLAY-NAME,PASSWORD,ROLE
7 2
8 3 * Usernames may contain lowercase letters or numbers and must be between 3 and 20 characters in length.
9 4 * Passwords must be between 6 and 30 characters in length.
10 5 * You can use RANDOM[n] for password to generate random n-digit password.
11 6 * ROLE must be one of these: 'admin', 'head_instructor', 'instructor', 'student'

```

- 13 Dengan contoh sebagai berikut:

Kode 2.22: Contoh kode untuk menambahkan pengguna

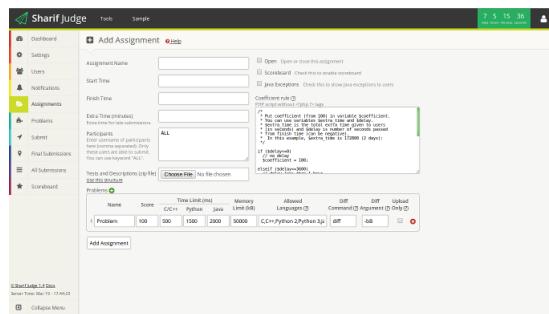
```

14
15 1 # This is a comment!
16 2 # This is another comment!
17 3 instructor,instructor@sharifjudge.ir,Instructor One,123456,head_instructor
18 4 instructor2,instructor2@sharifjudge.ir,Instructor Two,random[7],instructor
19 5 student1,st1@sharifjudge.ir,Student One,random[6],student
20 6 student2,st2@sharifjudge.ir,Student Two,random[6],student
21 7 student3,st3@sharifjudge.ir,Student Three,random[6],student
22 8 student4,st4@sharifjudge.ir,Student Four,random[6],student
23 9 student5,st5@sharifjudge.ir,Student Five,random[6],student
24 0 student6,st6@sharifjudge.ir,Student Six,random[6],student
25 1 student7,st7@sharifjudge.ir,Student Seven,random[6],student

```

27 2.2.5 Menambah *Assignment*

- 28 Pengguna dapat menambahkan *assignment* baru melalui bagian *Add* pada halaman *Assignmen-*
- 29 *ts*(dapat dilihat pada Gambar 2.2).



Gambar 2.2: Tampilan halaman *SharIF Judge* untuk menambahkan *assignment*

1 Berikut merupakan beberapa pengaturan pada halaman *Add Assignments*:

2 • ***Assignment Name***

3 *Assignment* akan ditampilkan sesuai dengan masukan pada daftar *assignment*.

4 • ***Start Time***

5 Pengguna tidak dapat mengumpulkan *assignment* sebelum waktu dimulai("Start Time").
6 Format pengaturan waktu untuk waktu mulai adalah MM/DD/YYYY HH:MM:SS dengan contoh
7 08/31/2013 12:00:00.

8 • ***Finish Time, Extra Time***

9 Pengguna tidak dapat mengumpulkan *assignment* setelah *Finish Time + Extra Time*. Pe-
10 nggumpulan *Assignment* pada *Extra Time* akan dikalikan seusai dengan koefisien. Pengguna
11 harus menulis skrip PHP untuk menghitung koefisien pada *field Coefficient Rule*. Format
12 pengaturan waktu untuk waktu selesai sama seperti waktu mulai yakni MM/DD/YYYY HH:MM:SS
13 dan format waktu tambahan menggunakan menit dengan contoh 120 (2 jam) atau 48*60 (2
14 hari).

15 • ***Participants***

16 Pengguna dapat memasukan *username* setiap partisipan atau menggunakan kata kunci *ALL*
17 untuk membiarkan seluruh pengguna melakukan pengumpulan. Contoh: *admin, instructor1 ,*
18 *instructor2 ,student1 , student2,student3 , student4*.

19 • ***Tests***

20 Pengguna dapat mengunggah *test case* dalam bentuk *zip file* sesuai dengan struktur pada
21 [2.2.7](#).

22 • ***Open***

23 Pengguna dapat membuka dan menutup *assigment* untuk pengguna *student* melalui pilihan
24 ini. Pengguna selain *student* tetap dapat mengumpulkan *assignment* apabila sudah ditutup.

1 • ***Score Board***

2 Pengguna dapat menghidupkan dan mematikan *score board* melalui pilihan ini.

3 • ***Java Exceptions***

4 Pengguna dapat menghidupkan dan mematikan fungsi untuk menunjukkan *java exceptions*
 5 kepada pengguna *students* dan tidak akan memengaruhi kode yang sudah di *judge* sebelumnya.
 6 Berikut merupakan tampilan apabila fitur *java exceptions* dinyalakan:

Kode 2.23: Contoh tampilan fitur *Java Exceptions*

```

7 1 Test 1
8 2 ACCEPT
9 3 Test 2
10 4 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
11 5 Test 3
12 6 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
13 7 Test 4
14 8 ACCEPT
15 9 Test 5
16 10 ACCEPT
17 11 Test 6
18 12 ACCEPT
19 20 Test 7
21 14 ACCEPT
22 15 Test 8
23 16 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
24 17 Test 9
25 18 Runtime Error (java.lang.StackOverflowError)
26 19 Test 10
27 20 Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
  
```

29 • ***Archived Assignment***

30 Pengguna dapat menghidupkan fitur ini dan *assignment* akan dibentuk dengan waktu selesai
 31 2038-01-18 00:00:00 (UTC + 7) dengan kata lain pengguna memiliki waktut tidak terhingga
 32 untuk mengumpulkan *assignment*.

33 • ***Coefficient Rule***

34 Pengguna dapat menuliskan skrip PHP pada bagian ini untuk menghitung koefisien dikalikan
 35 dengan skor. Pengguna harus memasukan koefisien (dari 100) dalam variabel `$coefficient`.
 36 Pengguna dapat menggunakan variabel `$extra_time` dan `$delay`. `$extra_time` merupakan
 37 total dari waktu tambahan yang diberikan kepada pengguna dalam detik dan `$delay` meru-
 38 pakan waktu dalam detik yang melewati waktu selesai(dapat berupa negatif). Skrip PHP
 39 pada bagian ini tidak boleh mengandung tag `<?php`, `<?`, dan `?>`. Berikut merupakan contoh
 40 skrip dimana `$extra_time` adalah 172800(2 hari):

Kode 2.24: Contoh skrip PHP

```

41 1 if ($delay<=0)
42 2 // no delay
43 3 $coefficient = 100;
44 4
45 5 elseif ($delay<=3600)
46 6 // delay less than 1 hour
47 7 $coefficient = ceil(100-((30*$delay)/3600));
48 8
49 9 elseif ($delay<=86400)
50 10 // delay more than 1 hour and less than 1 day
51 11 $coefficient = 70;
  
```

```

1   12
2   13 elseif (($delay-86400)<=3600)
3   14 // delay less than 1 hour in second day
4   15 $coefficient = ceil(70-((20*($delay-86400))/3600));
5   16
6   17 elseif (($delay-86400)<-86400)
7   18 // delay more than 1 hour in second day
8   19 $coefficient = 50;
9   20
10  21 elseif ($delay > $extra_time)
11  22 // too late
12  23 $coefficient = 0;
13

```

14 • ***Name***

15 Merupakan nama dari masalah pada *assignments*.

16 • ***Score***

17 Merupakan skor dari masalah pada *assignments*.

18 • ***Time Limit***

19 Pengguna dapat menentukan batas waktu untuk menjalankan kode dalam satuan milidetik.
20 Bahasa *Python* dan *Java* biasanya memiliki waktu lebih lambat dari *C/C++* sehingga
21 membutuhkan waktu lebih lama.

22 • ***Memory Limit***

23 Pengguna dapat menentukan batas memori dalam *kilobytes* namun, pengguna pembatasan
24 memori tidak terlalu akurat.

25 • ***Allowed Languages***

26 Pengguna dapat menentukan bahasa setiap masalah pada *assignment*(dipisahkan oleh koma).
27 Terdapat beberapa bahasa yang tersedia yaitu *C*, *C++*, *Java*, *Python 2*, *Python 3*, *Zip*,
28 *PDF*, dan *TXT*. Pengguna dapat memakai *Zip*, *PDF*, dan *TXT* apabila opsi *Upload Only*
29 dinyalakan. Contoh : *C*, *C++*, *Zip* atau *Python 2,Python 3*.

30 • ***Diff Command***

31 *Diff Command* digunakan untuk membandingkan keluaran dengan keluaran yang benar.
32 Secara asali, *SharIF Judge* menggunakan *diff* namun, pengguna dapat menggantinya pada
33 bagian ini dan bagian ini tidak boleh mengandung spasi.

34 • ***Diff Arguments***

35 Pengguna dapat mengatur argumen untuk *diff arguments* pada bagian ini. Pengguna dapat
36 melihat *man diff* untuk daftar lengkap argumen *diff*. *SharIF Judge* terdapat dua buah opsi
37 baru yakni *ignore* dan *identical*.

- 38 – *ignore* : *SharIF Judge* mengabaikan semua baris baru dan spasi.

- 1 – **identical** : *SharIF Judge* tidak mengabaikan apapun namun, keluaran dari *file* yang
 2 dikumpulkan harus identik dengan *test case* agar dapat diterima.

3 • ***Upload Only***

4 Pengguna dapat menghidupkan *Upload only* namun, *SharIF Judge* tidak akan menilai masalah
 5 tersebut. Pengguna dapat memakai *ZIP*, *PDF*, dan *TXT* pada *allowed languages* apabila
 6 pengguna menghidupkan bagian ini.

7 **2.2.6 Sample Assignment**

8 Berikut merupakan contoh dari *assignment* untuk melakukan pengujian *SharIF Judge*. Penambahan
 9 *Assignment* dapat dilakukan dengan memencet tombol *Add* pada halaman *Assignment*.

10 **Problems**

- 11 1. *Problem 1 (Sum)*: Program pengguna dapat membaca *integer n*, membaca *n integers* dan
 12 mengeluarkan hasil dari *integer* tersebut.

Sample Input	Sample Output
5	
53 78 0 4 9	145

- 13 2. *Problem 2 (Max)*: Program pengguna dapat membaca *integer n*, membaca *n integer*, dan
 14 mengeluarkan total dari dua buah *integer* terbesar diantara *n integer*.

Sample Input	Sample Output
7 162 173 159 164 181 158 175	356

- 15 3. *Problem 3 (Upload)*: Pengguna dapat mengunggah *file c* dan *zip* dan *problem* ini tidak akan
 16 dinilai karena hanya berupa *Upload Only*.

17 **Tests**

18 Pengguna dapat menemukan *file zip* pada direktori *Assignments*. Berikut merupakan susunan
 19 pohon dari ketiga *problems* diatas:

```

20 .
21 .
22 .|-- p1
23 .| |-- in
24 .| | |-- input1.txt
25 .| | |-- input2.txt
26 .| | |-- input3.txt
27 .| | |-- input4.txt
28 .| | |-- input5.txt
29 .| | |-- input6.txt
30 .| | |-- input7.txt
31 .| | |-- input8.txt
32 .| | |-- input9.txt
33 .| | |--- input10.txt
34 .| |-- out
35 .| | |--- output1.txt
36 .| | |--- tester.cpp
37 .| | |--- desc.md
38 |-- p2

```

```

19 |   |-- in
20 |   |   |-- input1.txt
21 |   |   |-- input2.txt
22 |   |   |-- input3.txt
23 |   |   |-- input4.txt
24 |   |   |-- input5.txt
25 |   |   |-- input6.txt
26 |   |   |-- input7.txt
27 |   |   |-- input8.txt
28 |   |   |-- input9.txt
29 |   |   --- input10.txt
30 |   |-- out
31 |   |   |-- output1.txt
32 |   |   |-- output2.txt
33 |   |   |-- output3.txt
34 |   |   |-- output4.txt
35 |   |   |-- output5.txt
36 |   |   |-- output6.txt
37 |   |   |-- output7.txt
38 |   |   |-- output8.txt
39 |   |   |-- output9.txt
40 |   |   --- output10.txt
41 |   |-- desc.md
42 |   --- Problem2.pdf
43 |   |-- p3
44 |   |   --- desc.md
45 |   --- SampleAssignment.pdf

```

29 *Problem 1* menggunakan metode "*Tester*" untuk mengecek hasil keluaran, sehingga memiliki
 30 file *tester.cpp*(*Tester Script*). *Problem 2* menggunakan metode "*Output Comparison*" untuk
 31 mengecek hasil keluaran, sehingga memiliki dua buah direktori *in* dan *out* yang berisi *test case*.
 32 *Problem 3* merupakan *problem "Upload-Only"*.

33 ***Sample Solutions***

34 Berikut merupakan *sample solutions* untuk *problem 1*:

35 Contoh solusi untuk bahasa *C*

Kode 2.25: Contoh skrip PHP

```

36
37 1 #include<stdio.h>
38 2 int main(){
39 3     int n;
40 4     scanf("%d",&n);
41 5     int i;
42 6     int sum =0 ;
43 7     int k;
44 8     for(i=0 ; i<n ; i++){
45 9         scanf("%d",&k);
46 0         sum+=k;
47 1     }
48 2     printf("%d\n",sum);
49 3     return 0;
50 4 }

```

52 Contoh solusi untuk bahasa *C++*

```

53
54 1 #include<stdio.h>
55 2 int main(){
56 3     int n;
57 4     scanf("%d",&n);
58 5     int i;
59 6     int sum =0 ;
60 7     int k;
61 8     for(i=0 ; i<n ; i++){
62 9         scanf("%d",&k);
63 0         sum+=k;
64 1     }
65 2     printf("%d\n",sum);
66 3     return 0;
67 4 }

```

69 Contoh solusi untuk bahasa *C*

```

21 import java.util.Scanner;
22 class sum {
23 {
24     public static void main(String[] args)
25     {
26         Scanner sc = new Scanner(System.in);
27         int n = sc.nextInt();
28         int sum =0;
29         for (int i=0 ; i<n ; i++)
30         {
31             int a = sc.nextInt();
32             sum += a;
33         }
34         System.out.println(sum);
35     }
36 }
```

19 Berikut merupakan contoh solusi untuk *problem 2*:

20 Contoh solusi untuk bahasa *C++*

```

22 #include<stdio.h>
23 int main(){
24     int n , m1=0, m2=0;
25     scanf("%d",&n);
26     for(;n--;){
27         int k;
28         scanf("%d",&k);
29         if(k>=m1){
30             m2=m1;
31             m1=k;
32         }
33         else if(k>m2)
34             m2=k;
35     }
36     printf("%d",m1+m2);
37     return 0;
38 }
```

40 contoh solusi untuk bahasa *C++*

```

42 #include<iostream>
43 using namespace std;
44 int main(){
45     int n , m1=0, m2=0;
46     cin >> n;
47     for(;n--;){
48         int k;
49         cin >> k;
50         if(k>=m1){
51             m2=m1;
52             m1=k;
53         }
54         else if(k>m2)
55             m2=k;
56     }
57     cout << (m1+m2) << endl ;
58     return 0;
59 }
```

61 2.2.7 Test Structure

62 Penambahan *assignment* harus disertakan dengan *file zip* berisikan *test cases*. *File zip* ini sebaiknya
63 berisikan *folder* untuk setiap masalah dengan nama *p1,p1* dan *p3* selain masalah *Upload-Only*.

64 Metode Pemeriksaan

65 Terdapat dua buah metode untuk melakukan pemeriksaan yakni metode *Input Output* dan metode
66 *Tester*.

67 1. Metode perbandingan *Input Output*

1 Pada metode ini, pengguna harus memberi masukan dan keluaran pada *folder problem*. Perangkat lunak akan memberikan *file test input* ke kode pengguna dan melakukan perbandingan dengan hasil keluaran kode pengguna. *File input* harus berada didalam *folder in* dengan nama *input1.txt, input2.txt, dst.* *File output* harus berada di dalam *folder out* dengan nama *output1.txt, output2.txt*.

2. Metode perbandingan *Tester*

7 Pada metode ini, pengguna harus menyediakan *file input test*, sebuah *file C++*, dan *file output test*(opsional). Perangkat lunak akan memberikan *file input test* ke kode pengguna dan mengambil keluaran pengguna. Selanjutnya, *tester.cpp* akan mengambil masukan pengguna, keluaran tes dan keluaran program pengguna. Jika keluaran pengguna benar maka perangkat lunak akan mengembalikan 1 sedangkan apabila salah maka perangkat lunak akan mengembalikan 0. Berikut adalah templat yang dapat digunakan pengguna untuk menuliskan *file tester.cpp*:

Kode 2.26: Templat kode *tester.cpp*

```

14 /*
15  * tester.cpp
16  */
17
18
19 #include <iostream>
20 #include <fstream>
21 #include <string>
22 using namespace std;
23 int main(int argc, char const *argv[])
24 {
25
26     ifstream test_in(argv[1]); /* This stream reads from test's input file */
27     ifstream test_out(argv[2]); /* This stream reads from test's output file */
28     ifstream user_out(argv[3]); /* This stream reads from user's output file */
29
30     /* Your code here */
31     /* If user's output is correct, return 0, otherwise return 1 */
32
33     ...
34
35 }
36
37 }
```

Sample File

Pengguna dapat menemukan *file sample test* pada direktori *assignments*. Berikut merupakan contoh dari pohon *file* tersebut:

```

40 .
41 |-- p1
42 |   |-- in
43 |   |   |-- input1.txt
44 |   |   |-- input2.txt
45 |   |   |-- input3.txt
46 |   |   |-- input4.txt
47 |   |   |-- input5.txt
48 |   |   |-- input6.txt
49 |   |   |-- input7.txt
50 |   |   |-- input8.txt
51 |   |   |-- input9.txt
52 |   |   |-- input10.txt
53 |   |   --- output1.txt
54 |   |-- out
55 |   |   --- output1.txt
56 |   |-- tester.cpp
57 |-- p2
58 |   |-- in
59 |   |   |-- input1.txt
60 |   |   |-- input2.txt
61 |   |   |-- input3.txt
62 |   |   |-- input4.txt
63 |   |   |-- input5.txt
```

```

1  24 |   |   |-- input6.txt
2  25 |   |   |-- input7.txt
3  26 |   |   |-- input8.txt
4  27 |   |   |-- input9.txt
5  28 |   |   --- input10.txt
6  29 |   |-- out
7  30 |   |   |-- output1.txt
8  31 |   |   |-- output2.txt
9  32 |   |   |-- output3.txt
10 33 |   |   |-- output4.txt
11 34 |   |   |-- output5.txt
12 35 |   |   |-- output6.txt
13 36 |   |   |-- output7.txt
14 37 |   |   |-- output8.txt
15 38 |   |   |-- output9.txt
16 39 |   |   --- output10.txt

```

18 Problem 1 menggunakan metode perbandingan *Tester*, sehingga memiliki file *tester.cpp*.
19 Berikut merupakan file untuk problem 1:

Kode 2.27: Kode metode perbandingan *tester* dengan bahasa *tester.cpp*

```

20 /*
21 * tester.cpp
22 */
23
24
25 #include <iostream>
26 #include <fstream>
27 #include <string>
28 using namespace std;
29 int main(int argc, char const *argv[])
30 {
31
32     ifstream test_in(argv[1]); /* This stream reads from test's input file */
33     ifstream test_out(argv[2]); /* This stream reads from test's output file */
34     ifstream user_out(argv[3]); /* This stream reads from user's output file */
35
36     /* Your code here */
37     /* If user's output is correct, return 0, otherwise return 1 */
38     /* e.g.: Here the problem is: read n numbers and print their sum: */
39
40     int sum, user_output;
41     user_out >> user_output;
42
43     if (test_out.good() ) // if test's output file exists
44     {
45         test_out >> sum;
46     }
47     else
48     {
49         int n, a;
50         sum=0;
51         test_in >> n;
52         for (int i=0 ; i<n ; i++){
53             test_in >> a;
54             sum += a;
55         }
56     }
57
58     if (sum == user_output)
59         return 0;
60     else
61         return 1;
62
63 }

```

65 Problem 2 menggunakan metode perbandingan *Input Output* sehingga memiliki folder *in* dan
66 folder *out* berisikan *test case*. Sedangkan problem 3 merupakan *Upload Only*, sehingga tidak
67 memiliki folder apapun.

68 2.2.8 Deteksi Kecurangan

69 SharIF Judge menggunakan *Moss* untuk mendeteksi kode yang mirip. *Moss(Measure of Software*
70 *Similarity*) merupakan sistem otomatis untuk menentukan kesamaan atau kemiripan dalam program.

- 1 Penggunaan utama *Moss* adalah untuk melakukan pemeriksaan plagiarisme pada kelas *programming*.
 2 Pengguna dapat mengirim hasil kode terakhir(*Final Submission*) ke *server Moss* dengan satu klik.
 3 Penggunaan *Moss* memiliki beberapa hal yang harus diatur oleh pengguna yakni:
 4 1. Pengguna harus mendapatkan *Moss User id* dan melakukan pengaturan pada *SharIF Judge*.
 5 Untuk mendapatkan *Moss User id*, pengguna dapat mendaftar pada halaman <http://theory.stanford.edu/~aiken/moss/>. Pengguna selanjutkan akan mendapatkan *email* berupa skrip
 6 *perl* berisikan *user id* pengguna. Berikut merupakan contoh dari potongan skrip *perl*:

Kode 2.28: Contoh potongan skrip *perl*

```

8
9
10
11
12 $server = 'moss.stanford.edu';
13 $port = '7690';
14 $noreq = "Request_not_sent.";
15 $usage = "Usage: moss [-x] [-l language] [-d] [-b basefile1] ... [-b basefileN] [-m#] [-c \"string\"] file1 file2 file3 ...
16     ...";
17
18 #
19 # The userid is used to authenticate your queries to the server; don't change it!
20 #
21 #
22 $userid=YOUR_MOSS_USER_ID;
23
24 #
25 # Process the command line options. This is done in a non-standard
26 # way to allow multiple -b's.
27 #
28 $opt_l = "c"; # default language is c
29 $opt_m = 10;
30 $opt_d = 0;
31 ...
32

```

33 *User id* pada skrip *perl* diatas dapat digunakan pada *SharIF Judge* untuk mendeteksi kecurangan. Pengguna dapat menyimpan *user id* pada halaman *SharIF Judge Moss* dan *SharIF Judge* akan menggunakan *user id* tersebut.

- 34 2. *Server* pengguna harus memiliki instalasi *perl* untuk menggunakan *Moss*.
 35 3. Pengguna dianjurkan untuk mendeteksi kode yang mirip setelah *assignment* selesai karena
 36 *SharIF Judge* akan mengirimkan hasil akhir kepada *Moss* dan pengguna(*students*) dapat
 37 mengganti hasil akhir sebelum *assignment* selesai.

40 2.2.9 Keamanan

41 Berikut merupakan langkah untuk melakukan pengaturan keamanan *SharIF Judge*:

42 1. Menggunakan *Sandbox*

43 Pengguna harus memastikan untuk menggunakan *sandbox* untuk bahasa *C/C++* dan menyalakan *Java Security Manager (Java Policy)* untuk bahasa *java*. Untuk menggunakan *sandbox*
 44 dapat dilihat pada sub bab [2.2.10](#).

46 2. Menggunakan *Shield*

47 Pengguna harus memastikan untuk menggunakan *shield* untuk bahasa *C*, *C++*, dan *Python*.
 48 Penggunaan *shield* dapat dilihat pada subbab [2.2.11](#).

3. Menjalankan sebagai *Non-Priviledge User*

Pengguna diwajibkan menjalankan kode yang telah dikumpulkan sebagai *non-priviledge user*. *Non-Priviledge User* adalah *user* yang tidak memiliki akses jaringan, tidak dapat menulis *file* apapun, dan tidak dapat membangun banyak proses.

Diasumsikan bahwa PHP dijalankan sebagai pengguna **www-data** pada server. Membangun *user* baru **restricted-user** dan melakukan pengaturan *password*. Selanjutnya, jalankan **sudo visudo** dan tambahkan kode **www-data ALL=(restricted_user) NOPASSWD: ALL** pada baris terakhir *file sudoers*.

- Pada *file tester\runcode.sh* ubah kode:

Kode 2.29: Kode *runcode.sh* awal

```
10
11 if $TIMEOUT_EXISTS; then
12     timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>err
13 else
14     $CMD <$IN >out 2>err
15 fi
```

menjadi:

Kode 2.30: Kode *runcode.sh* awal

```
18
19 if $TIMEOUT_EXISTS; then
20     sudo -u restricted_user timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>err
21 else
22     sudo -u restricted_user $CMD <$IN >out 2>err
23 fi
```

dan *uncomment* baris berikut:

Kode 2.31: Kode *runcode.sh* awal

```
26
27 sudo -u restricted_user pkill -9 -u restricted_user
```

- Mematikan akses jaringan untuk *restricted_user*

Pengguna dapat mematikan akses jaringan untuk *restricted_user* di *linux* menggunakan *iptables*. Setelah dimatikan lakukan pengujian menggunakan *ping* sebagai *restricted_user*.

- Menolak izin menulis

Pastikan tidak ada direktori atau *file* yang dapat ditulis oleh *restricted_user*.

- Membatasi jumlah proses

Pengguna dapat membatasi jumlah proses dari *restricted_user* dengan menambahkan kode dibawah melalui *file /etc/security/limits.conf*.

Kode 2.32: Contoh kode untuk membatasi jumlah proses

```
38
39 1 restricted_user    soft    nproc    3
40 2 restricted_user    hard    nproc    5
```

4. Menggunakan dua *server*

Pengguna dapat memakai dua *server* untuk antar muka web dan menangani permintaan web dan mengguna *server* lainnya untuk menjalankan kode yang sudah dikumpulkan. Penggunaan dua *server* mengurangi risiko menjalankan kode yang sudah dikumpulkan. Pengguna harus mengganti *source code SharIF Judge* untuk memakai hal ini.

2.2.10 Sandboxing

SharIF Judge menjalankan banyak *arbitrary codes* yang pengguna kumpulkan. *SharIF Judge* harus menjalankan kode pada lingkungan terbatas sehingga memerlukan perkakas untuk *sandbox* kode yang sudah dikumpulkan. Pengguna dapat meningkatkan keamanan dengan menghidupkan *shield* dan *Sandbox*.

C/C++ Sandboxing

SharIF Judge menggunakan *EasySandbox* untuk melakukan *sandboxing* kode C/C++. *EasySandbox* berguna untuk membatasi kode yang berjalan menggunakan *seccomp*. *Seccomp* merupakan mekanisme *sandbox* pada *Linux kernel*. Secara asali *EasySandbox* dimatikan pada *SharIF Judge* namun, pengguna dapat menghidupkannya melalui halaman *Settings*. Selain itu, pengguna juga harus "build *EasySandbox*" sebelum menyalakannya. Berikut merupakan cara melakukan *build EasySandbox*:

File *EasySandbox* terdapat pada *tester/easy sandbox*. Untuk membangun *EasySandbox* jalankan:

Kode 2.33: Kode *runcode.sh* awal

```
13 1 $ cd tester/easy sandbox
14 2 $ chmod +x runalltests.sh
15 3 $ chmod +x runtest.sh
16 4 $ make runtests
```

Jika keluaran berupa *message All test passed!* maka, *EasySandbox* berhasil dibangun dan dapat dinyalakan pada perangkat lunak.

Java Sandboxing

Secara asali, *Java Sandbox* sudah dinyalakan menggunakan *Java Security Manager*. Pengguna dapat menghidupkan atau mematikannya pada halaman *Settings*.

2.2.11 Shield

Shield merupakan mekanisme sangat simpel untuk mlarang jalannya kode yang berpotensi berbahaya. *Shield* bukan solusi *sandboxing* karena *shield* hanya menyediakan proteksi sebagian terhadap serangan remeh. Proteksi utama terhadap kode tidak terpercaya adalah dengan menghidupkan *Sandbox*(dapat dilihat pada subbab 2.2.10).

Shield untuk C/C++

Dengan menghidupkan *shield* untuk *c/c++*, *SharIF Judge* hanya perlu menambahkan `#define` pada awal kode yang dikumpulkan sebelum menjalankannya. Sebagai contoh, pengguna dapat mlarang penggunaan *goto* dengan menambahkan kode dibawah pada awal kode yang sudah dikumpulkan.

Kode 2.34: Kode *shield* untuk mlarang penggunaan *goto*

```
33 1 #define goto YouCannotUseGoto
```

Dengan kode diatas, semua kode yang menggunakan *goto* akan mendapatkan *compilation error*. Apabila pengguna menghidupkan *shield*, semua kode yang mengandung `#undef` akan mendapatkan *compilation error*.

- Menghidupkan *shield* untuk *C/C++*
Pengguna dapat menghidupkan atau mematikan *shield* pada halaman *settings*.
 - Menambahkan aturan untuk *C/C++* Daftar aturan `#define` untuk bahasa *C* terdapat pada halaman *tester/shield/defc.h* dan *tester/shield/defcpp.h* untuk bahasa *C++*. Pengguna dapat menambahkan aturan baru pada *file* tersebut pada halaman *settings*. Berikut merupakan contoh sintaks pada untuk menambahkan aturan :

Kode 2.35: Sintaks aturan `#define`

```
7 /*  
8  
9  
10 @file defc.h  
11 There should be a newline at end of this file.  
12 Put the message displayed to user after // in each line  
13  
14 e.g. If you want to disallow goto, add this line:  
15 #define goto errorNo13 //Goto is not allowed  
16  
17 */  
18  
19 #define system errorNo1      // "system" is not allowed  
20 #define freopen errorNo2    // File operation is not allowed  
21 #define fopen errorNo3      // File operation is not allowed  
22 #define printf errorNo4     // File operation is not allowed  
23 #define fscanf errorNo5     // File operation is not allowed  
24 #define feof errorNo6       // File operation is not allowed  
25 #define fclose errorNo7     // File operation is not allowed  
26 #define ifstream errorNo8   // File operation is not allowed  
27 #define ofstream errorNo9   // File operation is not allowed  
28 #define fork errorNo10      // Fork is not allowed  
29 #define clone errorNo11      // Clone is not allowed  
30 #define sleep errorNo12      // Sleep is not allowed
```

Pada akhir baris `file defc.h` dan `defcpp.h` harus terdapat baris baru. Terdapat banyak aturan yang tidak dapat digunakan pada `g++`, seperti aturan `#define fopen errorNo3` untuk bahasa `C++`.

35 *Shield* untuk *Python*

Penggunaan *shield* untuk *python* dapat dihidupkan melalui halaman *settings*. Dengan menghidupkan *shield* untuk *python*, *SharIF Judge* hanya menambahkan beberapa kode pada baris awal kode yang sudah dikumpulkan sebelum dijalankan. Penambahan kode digunakan untuk mencegah pemakaian fungsi berbahaya. Kode-kode tersebut terletak pada *file tester/shield/shield_py2.py* dan *tester/shield/shield_py3.py*. Berikut merupakan cara untuk keluar dari *shield* untuk *python* menggunakan fungsi yang telah di daftar hitamkan:

Kode 2.36: Cara keluar dari *shield* untuk *python*

```
42
43 1
44 2 # @file shield_py3.py
45 3
46 4 import sys
47 5 sys.modules['os']=None
48 6
49 7 BLACKLIST = [
50 8     #'__import__', # deny importing modules
51 9     'eval', # eval is evil
52 0     'open',
53 1     'file',
54 2     'exec',
55 3     'execfile',
56 4     'compile',
57 5     'reload',
58 6     #'input'
59 7 ]
60 8 for func in BLACKLIST:
```

```

19 if func in __builtins__._dict__:
20     del __builtins__._dict__[func]

```

4 2.3 CodeIgniter 4[3]

5 *CodeIgniter 4* merupakan versi terbaru dari *framework CodeIgniter* yang berfungsi untuk membantu
 6 pembentukan web. *CodeIgniter 4* dapat dipasang menggunakan *composer* ataupun dipasang secara
 7 manual dengan mengunduhnya dari situs web resmi. Berikut merupakan sintaks untuk melakukan
 8 pemasangan menggunakan *composer*.

```
9         composer create-project codeigniter4/appstarter project-root
```

10 Sintaks diatas akan mengunduh dan melakukan instalasi projek *CodeIgniter 4* dengan nama
 11 *project-root*. Sintaks *codeigniter4/appstarter* berfungsi untuk mengunduh aplikasi *skeleton* dari
 12 projek *CodeIgniter 4* yang berisikan kebutuhan data untuk melakukan *development* sebuah aplikasi.
 13 Setelah dilakukan pemasangan *CodeIgniter 4* memiliki lima buah direktori dengan struktur aplikasi
 14 sebagai berikut:

- 15 • **app/**

16 Direktori *app* berisikan semua kode yang dibutuhkan untuk menjalankan aplikasi web yang
 17 dibentuk. Direktori ini memiliki beberapa direktori didalamnya yaitu:

- 18 – *Config/* berfungsi dalam menyimpan *file* konfigurasi aplikasi web pengguna.
- 19 – *Controllers/* berfungsi sebagai penentu alur program yang dibentuk.
- 20 – *Database/* berfungsi sebagai penyimpan *file migrations* dan *seeds*.
- 21 – *Filters/* berfungsi dalam menyimpan *file* kelas *filter*.
- 22 – *Helpers/* berfungsi dalam menyimpan koleksi fungsi mandiri.
- 23 – *Language/* berfungsi sebagai tempat penyimpanan *string* dalam berbagai bahasa.
- 24 – *Libraries/* berfungsi dalam penyimpan kelas yang tidak termasuk kategori lain.
- 25 – *Models/* berfungsi untuk merepresentasikan data dari *database*.
- 26 – *ThirdParty/* *library ThirdParty* yang dapat digunakan pada aplikasi.
- 27 – *Views/* berisikan *file HTML* yang akan ditampilkan kepada pengguna.

- 28 • **public/**

29 Direktori *public* merupakan *web root* dari situs dan berisikan data-data yang dapat diakses
 30 oleh pengguna melalui *browser*. Direktori ini berisikan *file .htaccess*, *index.php*, dan *assets*
 31 dari aplikasi web yang dibentuk seperti gambar, *CSS*, ataupun *JavaScript*.

- 32 • **writable/**

33 Direktori *writable* berisikan data-data yang mungkin perlu ditulis seperti *file cache, logs*, dan
 34 *uploads*. Pengguna dapat menambahkan direktori baru sesuai dengan kebutuhan sehingga
 35 menambahkan keamanan pada direktori utama.

- 36 • **tests/**

37 Direktori ini menyimpan *file test* dan tidak perlu dipindahkan ke *server produksi*. Direktori
 38 *_support* berisikan berbagai jenis kelas *mock* dan keperluan yang dapat dipakai pengguna
 39 dalam menulis *tests*.

- 40 • **vendor/ atau system/**

1 Direktori ini berisikan *file* yang diperlukan dalam menjalani *framework* dan tidak boleh
 2 dimodifikasi oleh pengguna. Pengguna dapat melakukan *extend* atau membangun kelas baru
 3 untuk menambahkan fungsi yang diperlukan.

4 2.3.1 Models- Views- Controllers

5 *CodeIgniter 4* menggunakan pola *MVC* untuk mengatur *file* agar lebih simpel dalam menemukan
 6 *file* yang diperlukan. *MVC* menyimpan data, presentasi, dan alur program dalam *file* yang berbeda.

7 **Models**

8 *Models* berfungsi dalam menyimpan dan mengambil data dari tabel spesifik pada *database*. Data
 9 tersebut dapat berupa pengguna, pemberitahuan blog, transaksi, dll. *Models* pada umumnya
 10 disimpan pada direktori **app/Models** dan memiliki *namespace* sesuai dengan lokasi dari *file* tersebut.
 11 Kode 2.37 merupakan contoh dari *models*.

Kode 2.37: Contoh *Models*

```
12
13
14 <?php
15
16 namespace App\Models;
17
18 use CodeIgniter\Model;
19
20 class UserModel extends Model
21 {
22     protected $table      = 'users';
23     protected $primaryKey = 'id';
24
25     protected $useAutoIncrement = true;
26
27     protected $returnType   = 'array';
28     protected $useSoftDeletes = true;
29
30     protected $allowedFields = ['name', 'email'];
31
32     // Dates
33     protected $useTimestamps = false;
34     protected $dateFormat    = 'datetime';
35     protected $createdField  = 'created_at';
36     protected $updatedField  = 'updated_at';
37     protected $deletedField = 'deleted_at';
38
39     // Validation
40     protected $validationRules   = [];
41     protected $validationMessages = [];
42     protected $skipValidation   = false;
43     protected $cleanValidationRules = true;
44 }
45 }
```

47 Kode 2.37 merupakan contoh *Models* yang dapat digunakan pada *controllers*. *Models* tersebut
 48 terkoneksi dengan tabel **users** dengan *primarykey* **id**. *Model* pada *CodeIgniter 4* juga dapat
 49 digunakan untuk mencari, menyimpan, dan menghapus data untuk setiap tabel spesifik. Kode 2.38
 50 merupakan contoh penggunaan *model* untuk mencari data spesifik.

Kode 2.38: Contoh penggunaan *model* untuk mencari data spesifik

```
51
52 <?php
53
54 $users = $userModel->where('active', 1)->findAll();
```

56 Kode 2.38 menggabungkan *query* dengan metode pencarian *model* untuk mencari seluruh data
 57 **active**.

1 *Views*

- 2 *Views* merupakan halaman berisikan *HTML* dan sedikit PHP yang ditampilkan kepada pengguna
 3 ataupun dapat berupa pecahan halaman seperti *header*, *footer*, ataupun *sidebar*. *Views* biasanya
 4 terdapat pada `app/Views` dan mendapatkan data berupa variabel dari *controller* untuk ditampilkan.

Kode 2.39: Contoh *Views*

```

5 61 <html>
6 62   <head>
7 63     <title>My Blog</title>
8 64   </head>
9 65   <body>
10 66     <h1>Welcome to my Blog!</h1>
11 67   </body>
12 68 </html>
13 69
14 70
15 71
16 72
17 73
18 74
19 75
20 76
21 77
22 78
23 79
24 80
25 81
26 82
27 83
28 84
29 85
30 86
31 87
32 88
33 89
34 90
35 91
36 92
37 93
38 94
39 95
40 96
41 97
42 98
43 99
44 100
45 101
46 102
47 103
48 104
49 105
50 106
51 107
52 108
53 109
54 110
55 111
56 112
57 113
58 114
59 115
60 116
61 117
62 118
63 119
64 120
65 121
66 122
67 123
68 124
69 125
70 126
71 127
72 128
73 129
74 130
75 131
76 132
77 133
78 134
79 135
80 136
81 137
82 138
83 139
84 140
85 141
86 142
87 143
88 144
89 145
90 146
91 147
92 148
93 149
94 150
95 151
96 152
97 153
98 154
99 155
100 156
101 157
102 158
103 159
104 160
105 161
106 162
107 163
108 164
109 165
110 166
111 167
112 168
113 169
114 170
115 171
116 172
117 173
118 174
119 175
120 176
121 177
122 178
123 179
124 180
125 181
126 182
127 183
128 184
129 185
130 186
131 187
132 188
133 189
134 190
135 191
136 192
137 193
138 194
139 195
140 196
141 197
142 198
143 199
144 200
145 201
146 202
147 203
148 204
149 205
150 206
151 207
152 208
153 209
154 210
155 211
156 212
157 213
158 214
159 215
160 216
161 217
162 218
163 219
164 220
165 221
166 222
167 223
168 224
169 225
170 226
171 227
172 228
173 229
174 230
175 231
176 232
177 233
178 234
179 235
180 236
181 237
182 238
183 239
184 240
185 241
186 242
187 243
188 244
189 245
190 246
191 247
192 248
193 249
194 250
195 251
196 252
197 253
198 254
199 255
200 256
201 257
202 258
203 259
204 260
205 261
206 262
207 263
208 264
209 265
210 266
211 267
212 268
213 269
214 270
215 271
216 272
217 273
218 274
219 275
220 276
221 277
222 278
223 279
224 280
225 281
226 282
227 283
228 284
229 285
230 286
231 287
232 288
233 289
234 290
235 291
236 292
237 293
238 294
239 295
240 296
241 297
242 298
243 299
244 300
245 301
246 302
247 303
248 304
249 305
250 306
251 307
252 308
253 309
254 310
255 311
256 312
257 313
258 314
259 315
260 316
261 317
262 318
263 319
264 320
265 321
266 322
267 323
268 324
269 325
270 326
271 327
272 328
273 329
274 330
275 331
276 332
277 333
278 334
279 335
280 336
281 337
282 338
283 339
284 340
285 341
286 342
287 343
288 344
289 345
290 346
291 347
292 348
293 349
294 350
295 351
296 352
297 353
298 354
299 355
300 356
301 357
302 358
303 359
304 360
305 361
306 362
307 363
308 364
309 365
310 366
311 367
312 368
313 369
314 370
315 371
316 372
317 373
318 374
319 375
320 376
321 377
322 378
323 379
324 380
325 381
326 382
327 383
328 384
329 385
330 386
331 387
332 388
333 389
334 390
335 391
336 392
337 393
338 394
339 395
340 396
341 397
342 398
343 399
344 400
345 401
346 402
347 403
348 404
349 405
350 406
351 407
352 408
353 409
354 410
355 411
356 412
357 413
358 414
359 415
360 416
361 417
362 418
363 419
364 420
365 421
366 422
367 423
368 424
369 425
370 426
371 427
372 428
373 429
374 430
375 431
376 432
377 433
378 434
379 435
380 436
381 437
382 438
383 439
384 440
385 441
386 442
387 443
388 444
389 445
390 446
391 447
392 448
393 449
394 450
395 451
396 452
397 453
398 454
399 455
400 456
401 457
402 458
403 459
404 460
405 461
406 462
407 463
408 464
409 465
410 466
411 467
412 468
413 469
414 470
415 471
416 472
417 473
418 474
419 475
420 476
421 477
422 478
423 479
424 480
425 481
426 482
427 483
428 484
429 485
430 486
431 487
432 488
433 489
434 490
435 491
436 492
437 493
438 494
439 495
440 496
441 497
442 498
443 499
444 500
445 501
446 502
447 503
448 504
449 505
450 506
451 507
452 508
453 509
454 510
455 511
456 512
457 513
458 514
459 515
460 516
461 517
462 518
463 519
464 520
465 521
466 522
467 523
468 524
469 525
470 526
471 527
472 528
473 529
474 530
475 531
476 532
477 533
478 534
479 535
480 536
481 537
482 538
483 539
484 540
485 541
486 542
487 543
488 544
489 545
490 546
491 547
492 548
493 549
494 550
495 551
496 552
497 553
498 554
499 555
500 556
501 557
502 558
503 559
504 560
505 561
506 562
507 563
508 564
509 565
510 566
511 567
512 568
513 569
514 570
515 571
516 572
517 573
518 574
519 575
520 576
521 577
522 578
523 579
524 580
525 581
526 582
527 583
528 584
529 585
530 586
531 587
532 588
533 589
534 590
535 591
536 592
537 593
538 594
539 595
540 596
541 597
542 598
543 599
544 600
545 601
546 602
547 603
548 604
549 605
550 606
551 607
552 608
553 609
554 610
555 611
556 612
557 613
558 614
559 615
560 616
561 617
562 618
563 619
564 620
565 621
566 622
567 623
568 624
569 625
570 626
571 627
572 628
573 629
574 630
575 631
576 632
577 633
578 634
579 635
580 636
581 637
582 638
583 639
584 640
585 641
586 642
587 643
588 644
589 645
590 646
591 647
592 648
593 649
594 650
595 651
596 652
597 653
598 654
599 655
500 656
501 657
502 658
503 659
504 660
505 661
506 662
507 663
508 664
509 665
510 666
511 667
512 668
513 669
514 670
515 671
516 672
517 673
518 674
519 675
520 676
521 677
522 678
523 679
524 680
525 681
526 682
527 683
528 684
529 685
530 686
531 687
532 688
533 689
534 690
535 691
536 692
537 693
538 694
539 695
540 696
541 697
542 698
543 699
544 700
545 701
546 702
547 703
548 704
549 705
550 706
551 707
552 708
553 709
554 710
555 711
556 712
557 713
558 714
559 715
560 716
561 717
562 718
563 719
564 720
565 721
566 722
567 723
568 724
569 725
570 726
571 727
572 728
573 729
574 730
575 731
576 732
577 733
578 734
579 735
580 736
581 737
582 738
583 739
584 740
585 741
586 742
587 743
588 744
589 745
590 746
591 747
592 748
593 749
594 750
595 751
596 752
597 753
598 754
599 755
500 756
501 757
502 758
503 759
504 760
505 761
506 762
507 763
508 764
509 765
510 766
511 767
512 768
513 769
514 770
515 771
516 772
517 773
518 774
519 775
520 776
521 777
522 778
523 779
524 780
525 781
526 782
527 783
528 784
529 785
520 786
521 787
522 788
523 789
524 790
525 791
526 792
527 793
528 794
529 795
530 796
531 797
532 798
533 799
534 800
535 801
536 802
537 803
538 804
539 805
530 806
531 807
532 808
533 809
534 810
535 811
536 812
537 813
538 814
539 815
540 816
541 817
542 818
543 819
544 820
545 821
546 822
547 823
548 824
549 825
550 826
551 827
552 828
553 829
554 830
555 831
556 832
557 833
558 834
559 835
550 836
551 837
552 838
553 839
554 840
555 841
556 842
557 843
558 844
559 845
560 846
561 847
562 848
563 849
564 850
565 851
566 852
567 853
568 854
569 855
570 856
571 857
572 858
573 859
574 860
575 861
576 862
577 863
578 864
579 865
580 866
581 867
582 868
583 869
584 870
585 871
586 872
587 873
588 874
589 875
590 876
591 877
592 878
593 879
594 880
595 881
596 882
597 883
598 884
599 885
500 886
501 887
502 888
503 889
504 890
505 891
506 892
507 893
508 894
509 895
510 896
511 897
512 898
513 899
514 900
515 901
516 902
517 903
518 904
519 905
520 906
521 907
522 908
523 909
524 910
525 911
526 912
527 913
528 914
529 915
530 916
531 917
532 918
533 919
534 920
535 921
536 922
537 923
538 924
539 925
540 926
541 927
542 928
543 929
544 930
545 931
546 932
547 933
548 934
549 935
550 936
551 937
552 938
553 939
554 940
555 941
556 942
557 943
558 944
559 945
560 946
561 947
562 948
563 949
564 950
565 951
566 952
567 953
568 954
569 955
570 956
571 957
572 958
573 959
574 960
575 961
576 962
577 963
578 964
579 965
580 966
581 967
582 968
583 969
584 970
585 971
586 972
587 973
588 974
589 975
580 976
581 977
582 978
583 979
584 980
585 981
586 982
587 983
588 984
589 985
590 986
591 987
592 988
593 989
594 990
595 991
596 992
597 993
598 994
599 995
500 996
501 997
502 998
503 999
504 1000
505 1001
506 1002
507 1003
508 1004
509 1005
510 1006
511 1007
512 1008
513 1009
514 1010
515 1011
516 1012
517 1013
518 1014
519 1015
520 1016
521 1017
522 1018
523 1019
524 1020
525 1021
526 1022
527 1023
528 1024
529 1025
530 1026
531 1027
532 1028
533 1029
534 1030
535 1031
536 1032
537 1033
538 1034
539 1035
540 1036
541 1037
542 1038
543 1039
544 1040
545 1041
546 1042
547 1043
548 1044
549 1045
550 1046
551 1047
552 1048
553 1049
554 1050
555 1051
556 1052
557 1053
558 1054
559 1055
560 1056
561 1057
562 1058
563 1059
564 1060
565 1061
566 1062
567 1063
568 1064
569 1065
570 1066
571 1067
572 1068
573 1069
574 1070
575 1071
576 1072
577 1073
578 1074
579 1075
580 1076
581 1077
582 1078
583 1079
584 1080
585 1081
586 1082
587 1083
588 1084
589 1085
590 1086
591 1087
592 1088
593 1089
594 1090
595 1091
596 1092
597 1093
598 1094
599 1095
500 1096
501 1097
502 1098
503 1099
504 1100
505 1101
506 1102
507 1103
508 1104
509 1105
510 1106
511 1107
512 1108
513 1109
514 1110
515 1111
516 1112
517 1113
518 1114
519 1115
520 1116
521 1117
522 1118
523 1119
524 1120
525 1121
526 1122
527 1123
528 1124
529 1125
530 1126
531 1127
532 1128
533 1129
534 1130
535 1131
536 1132
537 1133
538 1134
539 1135
540 1136
541 1137
542 1138
543 1139
544 1140
545 1141
546 1142
547 1143
548 1144
549 1145
550 1146
551 1147
552 1148
553 1149
554 1150
555 1151
556 1152
557 1153
558 1154
559 1155
560 1156
561 1157
562 1158
563 1159
564 1160
565 1161
566 1162
567 1163
568 1164
569 1165
570 1166
571 1167
572 1168
573 1169
574 1170
575 1171
576 1172
577 1173
578 1174
579 1175
580 1176
581 1177
582 1178
583 1179
584 1180
585 1181
586 1182
587 1183
588 1184
589 1185
590 1186
591 1187
592 1188
593 1189
594 1190
595 1191
596 1192
597 1193
598 1194
599 1195
500 1196
501 1197
502 1198
503 1199
504 1200
505 1201
506 1202
507 1203
508 1204
509 1205
510 1206
511 1207
512 1208
513 1209
514 1210
515 1211
516 1212
517 1213
518 1214
519 1215
520 1216
521 1217
522 1218
523 1219
524 1220
525 1221
526 1222
527 1223
528 1224
529 1225
530 1226
531 1227
532 1228
533 1229
534 1230
535 1231
536 1232
537 1233
538 1234
539 1235
530 1236
531 1237
532 1238
533 1239
534 1240
535 1241
536 1242
537 1243
538 1244
539 1245
540 1246
541 1247
542 1248
543 1249
544 1250
545 1251
546 1252
547 1253
548 1254
549 1255
540 1256
541 1257
542 1258
543 1259
544 1260
545 1261
546 1262
547 1263
548 1264
549 1265
540 1266
541 1267
542 1268
543 1269
544 1270
545 1271
546 1272
547 1273
548 1274
549 1275
540 1276
541 1277
542 1278
543 1279
544 1280
545 1281
546 1282
547 1283
548 1284
549 1285
540 1286
541 1287
542 1288
543 1289
544 1290
545 1291
546 1292
547 1293
548 1294
549 1295
540 1296
541 1297
542 1298
543 1299
544 1300
545 1301
546 1302
547 1303
548 1304
549 1305
540 1306
541 1307
542 1308
543 1309
544 1310
545 1311
546 1312
547 1313
548 1314
549 1315
540 1316
541 1317
542 1318
543 1319
544 1320
545 1321
546 1322
547 1323
548 1324
549 1325
540 1326
541 1327
542 1328
543 1329
544 1330
545 1331
546 1332
547 1333
548 1334
549 1335
540 1336
541 1337
542 1338
543 1339
544 1340
545 1341
546 1342
547 1343
548 1344
549 1345
540 1346
541 1347
542 1348
543 1349
544 1350
545 1351
546 1352
547 1353
548 1354
549 1355
540 1356
541 1357
542 1358
543 1359
544 1360
545 1361
546 1362
547 1363
548 1364
549 1365
540 1366
541 1367
542 1368
543 1369
544 1370
545 1371
546 1372
547 1373
548 1374
549 1375
540 1376
541 1377
5
```

1 Selain itu, *CodeIgniter 4* menyediakan fungsi bernama *Controller Filters* dan kelas *IncomingRequest*.
 2 *Controller Filter* memiliki berfungsi untuk membiarkan pengguna membangun sebuah kondisi
 3 sebelum ataupun sesudah *controller* dijalankan. Kode 2.42 merupakan contoh penggunaan *filters*.

Kode 2.42: Contoh *Controllers Filters* pada *CodeIgniter 4*

```

4
5 1 <?php
6 2
7 3 namespace App\Filters;
8 4
9 5 use CodeIgniter\Filters\FilterInterface;
10 6 use CodeIgniter\HTTP\RequestInterface;
11 7 use CodeIgniter\HTTP\ResponseInterface;
12 8 use Config\Database;
13 9
14 0 class MyFilter implements FilterInterface
15 1 {
16 2     public function before(RequestInterface $request, $arguments = null)
17 3     {
18 4         $session = \Config\Services::session();
19 5         $db = Database::connect();
20 6
21 7         if ( !$db->tableExists('sessions'))
22 8             return redirect()->to('/install');
23 9         if ( !$session->get('logged_in')) // if not logged in
24 0             return redirect()->to('/login');
25 1     }
26 2
27 3     public function after(RequestInterface $request, ResponseInterface $response, $arguments = null)
28 4     {
29 5         // Do something here
30 6     }
31 7 }

```

33 Kode 2.42 merupakan contoh kode yang akan melakukan pengecekan tabel ataupun *session*
 34 sebelum *controller* dijalankan. Kode ini akan disimpan pada direktori *app/Filters*. Selanjutnya
 35 pengguna perlu menambahkan konfigurasi *filter* pada *routes* seperti sintaks berikut.

```

36
37 $routes->get('/', 'Dashboard::index', ['filter' => 'check-install:dual,noreturn']);

```

38 Sintaks diatas akan melakukan pengecekan pada *controller Dashboard::index* sebelum dan
 39 setelah *controller* tersebut dijalankan. Selanjutnya kelas *IncomingRequest* menyediakan representasi
 40 *object-oriented* sebuah *HTTP request* dari *client* seperti *browser*. Berikut merupakan contoh
 41 *IncomingRequest* mengakses data yang telah dikirimkan oleh *client*.

Kode 2.43: Contoh mengakses data menggunakan *IncomingRequest*

```

42
43 1 <?php
44 2
45 3 namespace App\Controllers;
46 4
47 5 use CodeIgniter\Controller;
48 6
49 7 class UserController extends Controller
50 8 {
51 9     public function index()
52 0     {
53 1         if ($this->request->isAJAX()) {
54 2             $something = $this->request->getVar('foo');
55 3         }
56 4     }
57 5 }

```

59 Kode 2.43 merupakan contoh untuk mengakses data yang telah dikirimkan oleh *client*. Sintaks
 60 *isAJAX* berfungsi untuk melakukan pengecekan apakan data yang dikirimkan melalui *AJAX*.
 61 Sedangkan sintaks *getVar('foo')* berfungsi untuk mengambil data yang telah dikirimkan dengan
 62 nama *foo*. Selain itu pengguna juga dapat menginisiasi kelas ini menggunakan sintaks berikut.

```
1             $request = \Config\Services::request();
```

- 2 Sintaks diatas akan menginisiasi kelas *request* dan menyimpannya menuju variabel.

3 2.3.2 Autoloading Files

- 4 *CodeIgniter 4* menyediakan fitur *autoloader* yang dapat digunakan dengan sedikit konfigurasi.
 5 Fitur ini dapat menemukan *namespaced classes* individual yang menggunakan struktur direktori
 6 *autoloading* pada *PSR-4*. Fitur ini juga dapat digunakan bersamaan dengan *autoloader* lain
 7 seperti *composer*. Konfigurasi fitur ini dapat dilakukan pada direktori *app/Config/Autoload.php*.
 8 Direktori ini berisikan dua buah *array* utama yakni *classmap* dan *psr4*. Kode 2.44 merupakan
 9 contoh konfigurasi menggunakan *namespace PSR-4*.

Kode 2.44: Contoh konfigurasi menggunakan *namespace PSR-4*.

```
10 11 <?php
12 13 namespace Config;
14 15 use CodeIgniter\Config\AutoloadConfig;
16 17 class Autoload extends AutoloadConfig
18 19 {
20 21     // ...
22 23     public $psr4 = [
24 25         APP_NAMESPACE => APPPATH, // For custom app namespace
26 27         'Config'      => APPPATH . 'Config',
28 29     ];
29 30
31 32     // ...
33 34 }
```

- 28 Kode 2.44 merupakan contoh konfigurasi untuk melakukan *mapping* menuju direktori. *Key* dari setiap baris merupakan *namespace* itu sendiri sedangkan *value* merupakan *path* dari direktori. Pengguna dapat melakukan pengecekan terhadap konfigurasi *namespace* menggunakan sintaks dibawah.

```
31               php spark namespaces
```

- 32 Sintaks diatas dapat dijalankan melalui *command line* pada aplikasi. Konfigurasi selanjutnya menggunakan *classmap* yang melakukan *link* terhadap *third-party libraries* yang tidak memiliki *namespace*. Kode 2.45 merupakan contoh konfigurasi menggunakan *classmap*.

Kode 2.45: Contoh konfigurasi menggunakan *classmap*.

```
35 36 <?php
37 38 namespace Config;
39 40 use CodeIgniter\Config\AutoloadConfig;
40 41 class Autoload extends AutoloadConfig
41 42 {
42 43     // ...
43 44     public $classmap = [
44 45         'Markdown' => APPPATH . 'ThirdParty/markdown.php',
45 46     ];
46 47
47 48     // ...
48 49 }
```

- 52 Kode 2.45 merupakan contoh konfigurasi menggunakan *classmap*. *Key* setiap row merupakan nama kelas yang ingin ditemukan sedangkan *value* merupakan *path* dari kelas itu sendiri.

1 2.3.3 Configuration

- 2 Konfigurasi pada *CodeIgniter* terletak pada direktori `app/Config`. File ini tidak ditempatkan pada
 - 3 satu buah file melainkan setiap kelas yang membutuhkan konfigurasi memiliki file yang berbeda.
 - 4 Pengguna dapat mengakses file *configuration* dengan beberapa cara berikut merupakan cara-caranya.

Kode 2.46: Contoh mengakses file *configuration*.

```
5
6 1 <?php
7 2 // Creating new configuration object by hand
8 3 $config = new \Config\Pager();
9 4
10 5 // Get shared instance with config function
11 6 $config = config('Pager');
12 7
13 8 // Access config class with namespace
14 9 $config = config('Config\\Pager');
15 0 $config = config(\Config\Pager::class);
16 1
17 2 // Creating a new object with config function
18 3 $config = config('Pager', false);
```

- 20 Kode 2.46 merupakan beberapa contoh untuk mengakses file *configuration*. Pengguna dapat mengak-
21 ses *configuration* secara manual menggunakan sintaks **new**, pengguna dapat mengakses *configuration*
22 menggunakan fungsi *config* dengan sintaks **config('namafile')**, menggunakan *namespace*, dan
23 membentuk objek baru menggunakan fungsi *config*. Selanjutnya pengguna dapat mengakses properti
24 yang terdapat pada file *config* tersebut menggunakan sintaks berikut.

- Sintaks diatas akan mengambil properti `perPage` yang terdapat pada variabel `config` yang sudah diinisiasi. Selain menggunakan `config` yang terdapat pada `CodeIgniter`, pengguna dapat membentuk file `config` secara manual. Kode 2.47 merupakan contoh isi file `config` yang dibentuk manual.

Kode 2.47: Contoh pembentukan file *configuration*.

```
29
30 1 <?php
31 2
32 3 namespace Config;
33 4
34 5 use CodeIgniter\Config\BaseConfig;
35 6
36 7 class CustomClass extends BaseConfig
37 8 {
38 9     public $siteName = 'My Great Site';
39 0     public $siteEmail = 'webmaster@example.com';
40 1     // ...
41 2 }
```

- 43 Kode [2.47](#) merupakan isi dari file *config* yang dibentuk secara manual. File ini akan disimpan pada
44 direktori `app/Config` dengan nama kelas yang akan *extends BaseConfig*. Selain menggunakan file
45 *config*, pengguna juga dapat melakukan konfigurasi menggunakan variabel *environment*. Penggu-
46 naan file *environment* merupakan cara melakukan konfigurasi terbaik untuk saat ini dikarenakan
47 kemudahan untuk mengubah konfigurasi pada saat melakukan *deploy* aplikasi. Konfigurasi dapat
48 diubah tanpa harus mengubah kode. *CodeIgniter* menyediakan sebuah file bernama *dotenv* atau
49 yang akan disebut `.env` selanjutnya. `.env` merupakan sebuah file yang terletak pada akar aplikasi.
50 File ini dapat berisikan seluruh konfigurasi yang diperlukan oleh aplikasi. Berikut merupakan
51 contoh variabel yang disimpan pada file `.env`.

Kode 2.48: Contoh variabel yang disimpan pada file .env.

```

11| S3_BUCKET = dotenv
12| SECRET_KEY = super_secret_key
13| CI_ENVIRONMENT = development

```

5 Kode 2.48 merupakan contoh variabel yang disimpan pada file `.env`. Variabel dapat berisikan
 6 konfigurasi yang bersifat pribadi seperti *password* dan *API keys*. Namun, konfigurasi hanya bersifat
 7 sebagai pengganti data sehingga pengguna tidak dapat membentuk data baru melainkan hanya
 8 mengubah data yang sudah ada sebelumnya.

9 2.3.4 *CodeIgniter URLs*

10 *CodeIgniter 4* menggunakan pendekatan *segment-based* dibandingkan menggunakan *query-string*
 11 untuk menghasilkan *URL* sehingga ramah manusia dan mesin pencari. Berikut merupakan contoh
 12 *URL* yang dihasilkan *CodeIgniter 4*:

13 <https://www.example.com/ci-blog/blog/news/2022/10?page=2>

14 *CodeIgniter 4* menghasilkan *URL* seperti diatas dengan membaginya menjadi:

- *Base URL* merupakan *URL* dasar dari aplikasi web yang dibentuk yaitu

16 <https://www.example.com/ci-blog/>

17 • *URI Path* merupakan alamat yang dituju yaitu `/ci-blog/blog/news/2022/10`

18 • *Route* juga merupakan alamat yang dituju tanpa *URL* dasar yaitu `/blog/news/2022/10`

19 • *Query* merupakan hasil dari *query* yang ingin ditampilkan yaitu `page=2`

20 Secara asali *CodeIgniter 4* membangun *URL* dengan `index.php` namun, pengguna dapat
 21 menghapus `file index.php` pada *URL* yang dibentuk. Pengguna dapat menghapus `index.php`
 22 sesuai dengan *server* yang digunakan. Berikut merupakan contoh dua buah *server* yang biasanya
 23 dipakai:

24 *Apache Web Server*

25 Pengguna dapat *URL* melalui `file .htaccess` dengan menyalakan ekstensi `mod_rewrite`. Kode 2.49
 26 merupakan contoh `file .htaccess` untuk menghapus `index.php` pada *URL* yang dibentuk.

Kode 2.49: Contoh `file .htaccess` pada *Apache Web Server*.

```

27
28 1 RewriteEngine On
29 2 RewriteCond %{REQUEST_FILENAME} !-f
30 3 RewriteCond %{REQUEST_FILENAME} !-d
31 4 RewriteRule ^(.*)$ index.php/$1 [L]

```

33 *File* diatas memperlakukan semua *HTTP Request* selain dari direktori dan *file* yang ada sebagai
 34 permintaan.

35 *NGINX*

36 Pengguna dapat mengubah *URL* menggunakan `try_files` yang akan mencari *URI* dan mengirimkan
 37 permintaan pada *URL* yang ingin dihilangkan. Kode 2.50 merupakan contoh penggunaan `try_files`
 38 untuk menghapus `index.php` pada *URL*.

Kode 2.50: Contoh penggunaan `try-files`.

```

1  location / {
2     try_files $uri $uri/ /index.php$is_args$args;
3 }
```

2.3.5 URI Routing

CodeIgniter 4 menyediakan dua buah *routing* yakni:

Defined Route Routing

Pengguna dapat mendefinisikan *route* secara manual untuk *URL* yang lebih fleksibel. Kode 2.51 merupakan contoh *route* yang didefinisikan secara manual.

Kode 2.51: Contoh *route* yang didefinisikan secara manual

```

11 <?php
12
13 $routes->get('product/(:num)', 'Catalog::productLookup');
```

Kode 2.51 merupakan contoh penggunaan *route* untuk menuju kelas Catalog dengan metode `productLookup`. Pengguna juga dapat memakai beberapa *HTTP verb* seperti *GET, POST, PUT, etc.* Selain menulis secara individu, pengguna dapat melakukan *grouping* pada route seperti Kode.

Kode 2.52: Contoh *route* yang menggunakan *grouping* manual

```

19 <?php
20
21 $routes->group('admin', static function ($routes) {
22     $routes->get('users', 'Admin\Users::index');
23     $routes->get('blog', 'Admin\Blog::index');
24 });
25
```

Kode 2.52 merupakan contoh penggunaan *grouping* untuk *URI* `admin/users` dan `admin/blog`.

Auto Routing

Pengguna dapat mendefinisikan *route* secara otomatis melalui fitur *Auto Routing* apabila tidak terdapat *route*. Pengguna dapat menyalakan fitur ini pada `app\Config\Routes.php` dengan cara berikut:

```
32 $routes->setAutoRoute(true);
```

Pengguna perlu mengubah `$autoRoutesImproved` menjadi `true` pada file `app\Config\Feature.php`. Selain menggunakan *auto routing* baru, pengguna dapat menggunakan *Auto Routing (Legacy)* yang terdapat pada *CodeIgniter 3* dengan cara seperti berikut:

2.3.6 Database

CodeIgniter 4 menyediakan kelas *database* yang dapat menyimpan, memasukan, memperbarui, dan menghapus data pada *database* sesuai dengan konfigurasi. Pengguna dapat melakukan konfigurasi untuk *database* yang ingin dikoneksikan melalui direktori `app\Config\Database.php` atau `file .env`. Kode 2.53 merupakan contoh pada direktori `Database.php`.

Kode 2.53: Contoh konfigurasi database pada *CodeIgniter 4*.

```

1 21 <?php
2 32
3 43 namespace Config;
4 54
5 65 use CodeIgniter\Database\Config;
6 76
7 87 class Database extends Config
8 98 {
9 109     public $default = [
10 110         'DSN'      => '',
11 121         'hostname' => 'localhost',
12 132         'username' => 'root',
13 143         'password' => '',
14 154         'database' => 'database_name',
15 165         'DBDriver' => 'MySQLi',
16 176         'DBPrefix' => '',
17 187         'pConnect' => true,
18 198         'DBDebug' => true,
19 209         'charset' => 'utf8',
20 220         'DBCollat' => 'utf8_general_ci',
21 231         'swapPre' => '',
22 242         'encrypt' => false,
23 253         'compress' => false,
24 264         'strictOn' => false,
25 275         'failover' => [],
26 286         'port'     => 3306,
27 297     ];
28 308
29 319 // ...
30 330 }
31 340
32 350
33 360

```

33 Kode 2.53 merupakan contoh konfigurasi untuk database bernama *database_name* dengan
34 *username* root. Selain itu, konfigurasi juga dapat dilakukan pada file *.env* untuk mempermudah
35 dalam pengubahan pada saat melakukan *deploy*. Kode 2.54 merupakan contoh konfigurasi pada file
36 *.env*:

Kode 2.54: Contoh konfigurasi database pada file *.env*.

```

37 381 database.default.username = 'root';
38 392 database.default.password = '';
39 403 database.default.database = 'ci4';
40

```

42 Kode 2.54 akan menyimpan konfigurasi pada grup *default* dengan *username* berupa root, tanpa
43 menggunakan *password*, dan juga dengan nama *database* ci4. Selain untuk melakukan koneksi
44 *database*, kelas ini dapat digunakan untuk menambahkan, menghapus, dan memperbarui data
45 pada *database*. Berikut merupakan contoh penggunaan *query* pada *database*:

Kode 2.55: Contoh penggunaan *query* menggunakan konfigurasi pada *CodeIgniter 4*.

```

46 471 <?php
47 482
48 493 $builder = $db->table('users');
49 504 $builder->select('title, content, date');
50 515 $query = $builder->get();
51

```

53 Kode 2.55 merupakan contoh penggunaan *query* untuk mengambil data *title, content*, dan *date*
54 pada tabel *users*. *CodeIgniter 4* juga menyediakan fitur untuk membangun *database* melalui fitur
55 bernama *Database Forge*. Pengguna dapat membangun, mengubah, menghapus tabel dan juga
56 menambahkan *field* pada tabel tersebut. Kode 2.56 merupakan contoh pembentukan *database*.

Kode 2.56: Contoh pembentukan tabel melalui *database forge*.

```

57 581 <?php
58 592
59 603 $fields = [
60 614     'id' => [
61 625         'type'      => 'INT',
62 636         'constraint' => 5,
63

```

```

17     'unsigned'      => true,
28     'auto_increment' => true,
39   ],
40   'title' => [
51     'type'          => 'VARCHAR',
62     'constraint'   => '100',
73     'unique'        => true,
84   ],
95   'author' => [
106     'type'          => 'VARCHAR',
117     'constraint'   => 100,
128     'default'       => 'King of Town',
139   ],
140   'description' => [
151     'type'          => 'TEXT',
162     'null'          => true,
173   ],
184   'status' => [
195     'type'          => 'ENUM',
206     'constraint'   => ['publish', 'pending', 'draft'],
217     'default'       => 'pending',
228   ],
239 ];
240 $forge->addField($fields);
251 $forge->createTable('table_name');

```

27 Kode 2.56 merupakan contoh pembentukan *database* dengan tabel bernama *table_name* yang
28 berisikan beberapa *field*.

29 2.3.7 Library

30 *CodeIgniter 4* menyediakan berbagai *library* untuk membantu pengguna dalam pembentukan aplikasi
31 web. Berikut merupakan contoh *library* yang disediakan oleh *CodeIgniter 4*:

32 Kelas Email

33 *CodeIgniter* menyediakan kelas *email* dengan fitur sebagai berikut:

- 34 • Beberapa Protokol: *Mail*, *Sendmail*, dan *SMTP*
- 35 • Enkripsi *TLS* dan *SSL* untuk *SMTP*
- 36 • Beberapa Penerima
- 37 • *CC* dan *BCCs*
- 38 • *HTML* atau *email* teks biasa
- 39 • Lampiran
- 40 • Pembungkus kata
- 41 • Prioritas
- 42 • Mode *BCC Batch*, memisahkan daftar *email* besar menjadi beberapa *BCC* kecil.
- 43 • Alat *Debugging email*

44 Pengguna dapat melakukan konfigurasi pada file *app/Config/Email.php* untuk melakukan
45 pengiriman *email*. Kode 2.57 merupakan contoh konfigurasi preferensi *email* secara manual.

Kode 2.57: Contoh kode untuk melakukan konfigurasi *email*.

```

46
47 1 <?php
48 2
49 3 $config['protocol'] = 'sendmail';
50 4 $config['mailPath'] = '/usr/sbin/sendmail';
51 5 $config['charset'] = 'iso-8859-1';
52 6 $config['wordWrap'] = true;
53 7
54 8 $email->initialize($config);

```

- 1 Selain itu, pengguna dapat melakukan pengiriman *email* sesuai dengan kebutuhan. Kode 2.58
 2 merupakan contoh penggunaan kelas *email* untuk mengirim *email*.

Kode 2.58: Contoh kode untuk melakukan pengiriman *email*.

```

3 41 <?php
5 42
6 43 $email = \Config\Services::email();
7 44
8 45 $email->setFrom('your@example.com', 'Your Name');
9 46 $email->setTo('someone@example.com');
10 47 $email->setCC('another@another-example.com');
11 48 $email->setBCC('them@their-example.com');
12 49
13 50 $email->setSubject('Email Test');
14 51 $email->setMessage('Testing the email class.');
15 52
16 53 $email->send();

```

- 18 Kode 2.58 merupakan contoh penggunaan kelas *email* untuk mengirimim *email* dari `your@example.com`
 19 kepada `someone@example.com` dengan subjek `Email Test` dan pesan `Testing the email class`.

20 Session

- 21 *Library session* memperbolehkan aplikasi untuk melihat jejak dari aktivitas dari pengguna aplikasi
 22 tersebut. Data *session* dapat disimpan pada beberapa tempat yakni file, *database*, *memcached*,
 23 *redis*, dan *array*. *Session* dapat dilakukan inisiasi dengan sintaks sebagai berikut:

```
24 $session = \Config\Services::session($config);
```

- 25 Sintaks diatas akan melakukan inisiasi *library session* dan pengguna dapat memanggilnya menggu-
 26 nakan sintaks berikut:

```
27 $session
```

- 28 *Session* bekerja saat halaman dari aplikasi termuat. Aplikasi akan melakukan pengecekan terhadap
 29 *session cookie* dan melakukan pembaharuan sesuai dengan *session cookie* yang didapat. Pengguna
 30 juga dapat mengambil data yang tesimpan pada *session* menggunakan sintaks berikut:

```
31 $item = $session->get('item');
```

- 32 Sintaks diatas akan mengambil data *session* dengan nama *item* dan menyimpannya menuju sebuah
 33 variabel. Selain itu, *session* juga dapat dihapus menggunakan sintaks sebagai berikut:

```
34 $session->destroy();
```

- 35 Sintkas diatas akan membersihkan *session* dari pengguna seperti saat melakukan *logout*. Secara
 36 *default session* akan menyimpan data menuju file namun, pengguna dapat melakukan konfigurasi
 37 sesuai dengan keinginan. Berikut merupakan konfigurasi untuk menyimpan data *session* menuju
 38 *database*:

Kode 2.59: Contoh kode untuk konfigurasi penyimpanan *session*.

```

39 401 class Session extends BaseConfig
402 {
413 // ...
424 public string $driver = 'CodeIgniter\Session\Handlers\DatabaseHandler';
435
446 // ...
457 public string $savePath = 'ci_sessions';
468
479 // ...
480 }
```

- 1 Kode 2.59 akan mengubah penyimpanan menggunakan *databasehandler* dengan tabel penyimpanan
 2 bernama *ci_sessions*. Konfigurasi dilakukan pada direktori *app/Config*.

3 ***Working with Uploaded Files***

- 4 Pengunggahan *file* terdapat empat buah proses sebagai berikut:
 5 1. Dibentuk sebuah form untuk pengguna memilih dan mengunggah *file*.
 6 2. Setelah *file* diunggah, *file* akan dipindahkan menuju direktori yang dipilih.
 7 3. Pada pengiriman dan pemindahan *file* dilakukan validasi sesuai dengan ketentuan yang ada.
 8 4. Setelah *file* diterima akan dikeluarkan pesan berhasil.
 9 Perangkat lunak akan menerima *file* dari *views* yang nantinya akan dilakukan validasi pada *controller*.
 10 Kode 2.60 merupakan contoh *view* untuk melakukan pengunggahan *file*.

Kode 2.60: Contoh kode untuk melakukan pengunggahan *file*.

```

11 11<!DOCTYPE html>
12 12<html lang="en">
13 13<head>
14 14    <title>Upload Form</title>
15 15</head>
16 16<body>
17 17
18 18<?php foreach ($errors as $error): ?>
19 19    <li><?= esc($error) ?></li>
20 20<?php endforeach ?>
21 21
22 22<?= form_open_multipart('upload/upload') ?>
23 23    <input type="file" name="userfile" size="20">
24 24    <br><br>
25 25    <input type="submit" value="upload">
26 26</form>
27 27
28 28</body>
29 29</html>
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
62 62
63 63
64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 78
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88
89 89
90 90
91 91
92 92
93 93
94 94
95 95
96 96
97 97
98 98
99 99
100 100
101 101
102 102
103 103
104 104
105 105
106 106
107 107
108 108
109 109
110 110
111 111
112 112
113 113
114 114
115 115
116 116
117 117
118 118
119 119
120 120
121 121
122 122
123 123
124 124
125 125
126 126
127 127
128 128
129 129
130 130
131 131
132 132
133 133
134 134
135 135
136 136
137 137
138 138
139 139
140 140
141 141
142 142
143 143
144 144
145 145
146 146
147 147
148 148
149 149
150 150
151 151
152 152
153 153
154 154
155 155
156 156
157 157
158 158
159 159
160 160
161 161
162 162
163 163
164 164
165 165
166 166
167 167
168 168
169 169
170 170
171 171
172 172
173 173
174 174
175 175
176 176
177 177
178 178
179 179
180 180
181 181
182 182
183 183
184 184
185 185
186 186
187 187
188 188
189 189
190 190
191 191
192 192
193 193
194 194
195 195
196 196
197 197
198 198
199 199
200 200
201 201
202 202
203 203
204 204
205 205
206 206
207 207
208 208
209 209
210 210
211 211
212 212
213 213
214 214
215 215
216 216
217 217
218 218
219 219
220 220
221 221
222 222
223 223
224 224
225 225
226 226
227 227
228 228
229 229
230 230
231 231
232 232
233 233
234 234
235 235
236 236
237 237
238 238
239 239
240 240
241 241
242 242
243 243
244 244
245 245
246 246
247 247
248 248
249 249
250 250
251 251
252 252
253 253
254 254
255 255
256 256
257 257
258 258
259 259
260 260
261 261
262 262
263 263
264 264
265 265
266 266
267 267
268 268
269 269
270 270
271 271
272 272
273 273
274 274
275 275
276 276
277 277
278 278
279 279
280 280
281 281
282 282
283 283
284 284
285 285
286 286
287 287
288 288
289 289
290 290
291 291
292 292
293 293
294 294
295 295
296 296
297 297
298 298
299 299
300 300
301 301
302 302
303 303
304 304
305 305
306 306
307 307
308 308
309 309
310 310
311 311
312 312
313 313
314 314
315 315
316 316
317 317
318 318
319 319
320 320
321 321
322 322
323 323
324 324
325 325
326 326
327 327
328 328
329 329
330 330
331 331
332 332
333 333
334 334
335 335
336 336
337 337
338 338
339 339
340 340
341 341
342 342
343 343
344 344
345 345
346 346
347 347
348 348
349 349
350 350
351 351
352 352
353 353
354 354
355 355
356 356
357 357
358 358
359 359
360 360
361 361
362 362
363 363
364 364
365 365
366 366
367 367
368 368
369 369
370 370
371 371
372 372
373 373
374 374
375 375
376 376
377 377
378 378
379 379
380 380
381 381
382 382
383 383
384 384
385 385
386 386
387 387
388 388
389 389
390 390
391 391
392 392
393 393
394 394
395 395
396 396
397 397
398 398
399 399
400 400
401 401
402 402
403 403
404 404
405 405
406 406
407 407
408 408
409 409
410 410
411 411
412 412
413 413
414 414
415 415
416 416
417 417
418 418
419 419
420 420
421 421
422 422
423 423
424 424
425 425
426 426
427 427
428 428
429 429
430 430
431 431
432 432
433 433
434 434
435 435
436 436
437 437
438 438
439 439
440 440
441 441
442 442
443 443
444 444
445 445
446 446
447 447
448 448
449 449
450 450
451 451
452 452
453 453
454 454
455 455
456 456
457 457
458 458
459 459
460 460
461 461
462 462
463 463
464 464
465 465
466 466
467 467
468 468
469 469
470 470
471 471
472 472
473 473
474 474
475 475
476 476
477 477
478 478
479 479
480 480
481 481
482 482
483 483
484 484
485 485
486 486
487 487
488 488
489 489
490 490
491 491
492 492
493 493
494 494
495 495
496 496
497 497
498 498
499 499
500 500
501 501
502 502
503 503
504 504
505 505
506 506
507 507
508 508
509 509
510 510
511 511
512 512
513 513
514 514
515 515
516 516
517 517
518 518
519 519
520 520
521 521
522 522
523 523
524 524
525 525
526 526
527 527
528 528
529 529
530 530
531 531
532 532
533 533
534 534
535 535
536 536
537 537
538 538
539 539
540 540
541 541
542 542
543 543
544 544
545 545
546 546
547 547
548 548
549 549
550 550
551 551
552 552
553 553
554 554
555 555
556 556
557 557
558 558
559 559
560 560
561 561
562 562
563 563
564 564
565 565
566 566
567 567
568 568
569 569
570 570
571 571
572 572
573 573
574 574
575 575
576 576
577 577
578 578
579 579
580 580
581 581
582 582
583 583
584 584
585 585
586 586
587 587
588 588
589 589
590 590
591 591
592 592
593 593
594 594
595 595
596 596
597 597
598 598
599 599
600 600
601 601
602 602
603 603
604 604
605 605
606 606
607 607
608 608
609 609
610 610
611 611
612 612
613 613
614 614
615 615
616 616
617 617
618 618
619 619
620 620
621 621
622 622
623 623
624 624
625 625
626 626
627 627
628 628
629 629
630 630
631 631
632 632
633 633
634 634
635 635
636 636
637 637
638 638
639 639
640 640
641 641
642 642
643 643
644 644
645 645
646 646
647 647
648 648
649 649
650 650
651 651
652 652
653 653
654 654
655 655
656 656
657 657
658 658
659 659
660 660
661 661
662 662
663 663
664 664
665 665
666 666
667 667
668 668
669 669
670 670
671 671
672 672
673 673
674 674
675 675
676 676
677 677
678 678
679 679
680 680
681 681
682 682
683 683
684 684
685 685
686 686
687 687
688 688
689 689
690 690
691 691
692 692
693 693
694 694
695 695
696 696
697 697
698 698
699 699
700 700
701 701
702 702
703 703
704 704
705 705
706 706
707 707
708 708
709 709
710 710
711 711
712 712
713 713
714 714
715 715
716 716
717 717
718 718
719 719
720 720
721 721
722 722
723 723
724 724
725 725
726 726
727 727
728 728
729 729
730 730
731 731
732 732
733 733
734 734
735 735
736 736
737 737
738 738
739 739
740 740
741 741
742 742
743 743
744 744
745 745
746 746
747 747
748 748
749 749
750 750
751 751
752 752
753 753
754 754
755 755
756 756
757 757
758 758
759 759
760 760
761 761
762 762
763 763
764 764
765 765
766 766
767 767
768 768
769 769
770 770
771 771
772 772
773 773
774 774
775 775
776 776
777 777
778 778
779 779
780 780
781 781
782 782
783 783
784 784
785 785
786 786
787 787
788 788
789 789
790 790
791 791
792 792
793 793
794 794
795 795
796 796
797 797
798 798
799 799
800 800
801 801
802 802
803 803
804 804
805 805
806 806
807 807
808 808
809 809
810 810
811 811
812 812
813 813
814 814
815 815
816 816
817 817
818 818
819 819
820 820
821 821
822 822
823 823
824 824
825 825
826 826
827 827
828 828
829 829
830 830
831 831
832 832
833 833
834 834
835 835
836 836
837 837
838 838
839 839
840 840
841 841
842 842
843 843
844 844
845 845
846 846
847 847
848 848
849 849
850 850
851 851
852 852
853 853
854 854
855 855
856 856
857 857
858 858
859 859
860 860
861 861
862 862
863 863
864 864
865 865
866 866
867 867
868 868
869 869
870 870
871 871
872 872
873 873
874 874
875 875
876 876
877 877
878 878
879 879
880 880
881 881
882 882
883 883
884 884
885 885
886 886
887 887
888 888
889 889
890 890
891 891
892 892
893 893
894 894
895 895
896 896
897 897
898 898
899 899
900 900
901 901
902 902
903 903
904 904
905 905
906 906
907 907
908 908
909 909
910 910
911 911
912 912
913 913
914 914
915 915
916 916
917 917
918 918
919 919
920 920
921 921
922 922
923 923
924 924
925 925
926 926
927 927
928 928
929 929
930 930
931 931
932 932
933 933
934 934
935 935
936 936
937 937
938 938
939 939
940 940
941 941
942 942
943 943
944 944
945 945
946 946
947 947
948 948
949 949
950 950
951 951
952 952
953 953
954 954
955 955
956 956
957 957
958 958
959 959
960 960
961 961
962 962
963 963
964 964
965 965
966 966
967 967
968 968
969 969
970 970
971 971
972 972
973 973
974 974
975 975
976 976
977 977
978 978
979 979
980 980
981 981
982 982
983 983
984 984
985 985
986 986
987 987
988 988
989 989
990 990
991 991
992 992
993 993
994 994
995 995
996 996
997 997
998 998
999 999
1000 1000
1001 1001
1002 1002
1003 1003
1004 1004
1005 1005
1006 1006
1007 1007
1008 1008
1009 1009
1010 1010
1011 1011
1012 1012
1013 1013
1014 1014
1015 1015
1016 1016
1017 1017
1018 1018
1019 1019
1020 1020
1021 1021
1022 1022
1023 1023
1024 1024
1025 1025
1026 1026
1027 1027
1028 1028
1029 1029
1030 1030
1031 1031
1032 1032
1033 1033
1034 1034
1035 1035
1036 1036
1037 1037
1038 1038
1039 1039
1040 1040
1041 1041
1042 1042
1043 1043
1044 1044
1045 1045
1046 1046
1047 1047
1048 1048
1049 1049
1050 1050
1051 1051
1052 1052
1053 1053
1054 1054
1055 1055
1056 1056
1057 1057
1058 1058
1059 1059
1060 1060
1061 1061
1062 1062
1063 1063
1064 1064
1065 1065
1066 1066
1067 1067
1068 1068
1069 1069
1070 1070
1071 1071
1072 1072
1073 1073
1074 1074
1075 1075
1076 1076
1077 1077
1078 1078
1079 1079
1080 1080
1081 1081
1082 1082
1083 1083
1084 1084
1085 1085
1086 1086
1087 1087
1088 1088
1089 1089
1090 1090
1091 1091
1092 1092
1093 1093
1094 1094
1095 1095
1096 1096
1097 1097
1098 1098
1099 1099
1100 1100
1101 1101
1102 1102
1103 1103
1104 1104
1105 1105
1106 1106
1107 1107
1108 1108
1109 1109
1110 1110
1111 1111
1112 1112
1113 1113
1114 1114
1115 1115
1116 1116
1117 1117
1118 1118
1119 1119
1120 1120
1121 1121
1122 1122
1123 1123
1124 1124
1125 1125
1126 1126
1127 1127
1128 1128
1129 1129
1130 1130
1131 1131
1132 1132
1133 1133
1134 1134
1135 1135
1136 1136
1137 1137
1138 1138
1139 1139
1140 1140
1141 1141
1142 1142
1143 1143
1144 1144
1145 1145
1146 1146
1147 1147
1148 1148
1149 1149
1150 1150
1151 1151
1152 1152
1153 1153
1154 1154
1155 1155
1156 1156
1157 1157
1158 1158
1159 1159
1160 1160
1161 1161
1162 1162
1163 1163
1164 1164
1165 1165
1166 1166
1167 1167
1168 1168
1169 1169
1170 1170
1171 1171
1172 1172
1173 1173
1174 1174
1175 1175
1176 1176
1177 1177
1178 1178
1179 1179
1180 1180
1181 1181
1182 1182
1183 1183
1184 1184
1185 1185
1186 1186
1187 1187
1188 1188
1189 1189
1190 1190
1191 1191
1192 1192
1193 1193
1194 1194
1195 1195
1196 1196
1197 1197
1198 1198
1199 1199
1200 1200
1201 1201
1202 1202
1203 1203
1204 1204
1205 1205
1206 1206
1207 1207
1208 1208
1209 1209
1210 1210
1211 1211
1212 1212
1213 1213
1214 1214
1215 1215
1216 1216
1217 1217
1218 1218
1219 1219
1220 1220
1221 1221
1222 1222
1223 1223
1224 1224
1225 1225
1226 1226
1227 1227
1228 1228
1229 1229
1230 1230
1231 1231
1232 1232
1233 1233
1234 1234
1235 1235
1236 1236
1237 1237
1238 1238
1239 1239
1240 1240
1241 1241
1242 1242
1243 1243
1244 1244
1245 1245
1246 1246
1247 1247
1248 1248
1249 1249
1250 1250
1251 1251
1252 1252
1253 1253
1254 1254
1255 1255
1256 1256
1257 1257
1258 1258
1259 1259
1260 1260
1261 1261
1262 1262
1263 1263
1264 1264
1265 1265
1266 1266
1267 1267
1268 1268
1269 1269
1270 1270
1271 1271

```

```

27     ],
28 ],
29 ;
30 if (! $this->validate($validationRule)) {
31     $data = ['errors' => $this->validator->getErrors()];
32
33     return view('upload_form', $data);
34 }
35
36 $img = $this->request->getFile('userfile');
37
38 if (! $img->hasMoved()) {
39     $filepath = WRITEPATH . 'uploads/' . $img->store();
40
41     $data = ['uploaded_fileinfo' => new File($filepath)];
42
43     return view('upload_success', $data);
44 }
45
46 $data = ['errors' => 'The file has already been moved.'];
47
48 return view('upload_form', $data);
49 }
50 }
```

26 Kode 2.61 terdapat dua buah fungsi yaitu:

- 27 • `index()` yang mengembalikan *view* bernama `upload_form`
- 28 • `upload()` yang memberikan aturan untuk melakukan validasi dan melakukan penyimpanan
29 pada direktori `uploads`.

30 ***Working with URIs***

31 *CodeIgniter 4* menyediakan solusi berbasis *object oriented* untuk bekerja dengan *URI*. *Library*
32 ini simpel dan memastikan struktur selalu benar tanpa melihat *URI* yang kompleks. Berikut
33 merupakan sintaks untuk membentuk sebuah *URI*:

```
34 $uri = new \CodeIgniter\HTTP\URI('http://www.example.com/some/path');
```

35 Sintaks diatas akan membentuk *URI* dengan nama `example.com` dengan beberapa *segment*. *URI*
36 yang sudah dibentuk selanjutnya dapat diambil menggunakan sintaks berikut:

```
37 $uri = $this->request->getUri();
```

38 Sintaks diatas akan mengambil *URI* yang terdapat pada halaman tersebut. Selain dapat mengambil
39 *URI*, *library* ini juga dapat mengambil *segment* yang terdapat pada *URI*. Berikut merupakan sintaks
40 untuk mengambil *segment* yang terdapat pada *URI*.

```
41 $segments = $uri->getSegments();
```

42 Sintaks diatas akan mengambil *segment* yang tersedia pada *URI* dan menghasilkan *array* berisikan
43 *value* dari setiap *segment* yang tersedia.

44 ***Validation***

45 *CodeIgniter 4* menyediakan *library* untuk melakukan validasi terhadap data yang dikirimkan oleh
46 pengguna. Data yang divalidasi dapat diberikan aturan-aturan sesuai dengan konfigurasi pengguna.
47 Kode 2.62 merupakan contoh penggunaan *validation*.

Kode 2.62: Contoh kode untuk melakukan pengumpulan data.

```

1<html>
2<head>
3    <title>My Form</title>
4</head>
5<body>
6
7    <?= validation_list_errors() ?>
8
9    <?= form_open('form') ?>
10
11        <h5>Username</h5>
12        <input type="text" name="username" value="<?= set_value('username') ?>" size="50">
13
14        <h5>Password</h5>
15        <input type="text" name="password" value="<?= set_value('password') ?>" size="50">
16
17        <h5>Password Confirm</h5>
18        <input type="text" name="passconf" value="<?= set_value('passconf') ?>" size="50">
19
20        <h5>Email Address</h5>
21        <input type="text" name="email" value="<?= set_value('email') ?>" size="50">
22
23    <div><input type="submit" value="Submit"></div>
24
25    <?= form_close() ?>
26
27</body>
28</html>

```

31 Kode 2.62 merupakan contoh penggunaan *validation* pada halaman *view*. Pengambilan *error*
 32 dapat menggunakan sintaks `validation_list_errors()`. Selanjutnya akan digunakan fungsi
 33 `form_open` untuk membuka *tag form* sesuai dengan *url* yang sudah dibentuk. Setiap input akan
 34 diberikan *name* sesuai dengan aturan yang ingin diberikan. Setelah *tag form* selesai maka akan
 35 ditutup dengan sintaks `form_close`. Setelah itu data-data yang sudah dimasukan akan dikirimkan
 36 menuju *controller* seperti pada kode 2.63.

Kode 2.63: Contoh kode untuk melakukan validasi data yang sudah dikumpulkan.

```

37<?php
38
39 namespace App\Controllers;
40
41 class Form extends BaseController
42 {
43     protected $helpers = ['form'];
44
45     public function index()
46     {
47         if (! $this->request->is('post')) {
48             return view('signup');
49         }
50
51         $rules = [];
52
53         if (! $this->validate($rules)) {
54             return view('signup');
55         }
56
57         // If you want to get the validated data.
58         $validData = $this->validator->getValidated();
59
60         return view('success');
61     }
62 }

```

65 Data-data yang sudah diberikan oleh pengguna akan dijalankan menggunakan *controller* diatas.
 66 Sintaks `if (! $this->request->is('post'))` akan melakukan pengecekan apakah *request* yang
 67 diberikan berupa *post* atau tidak. Selanjutnya dapat ditentukan aturan pada variabel `rules` yang
 68 nantinya akan dilakukan validasi menggunakan fungsi `validate`. Fungsi `validate` akan mengecek
 69 data yang diberikan dan menentukan apakah sudah sesuai dengan aturan yang ada atau belum.

1 Kode 2.64 merupakan contoh pembentukan aturan sesuai dengan nama *form* yang ada.

Kode 2.64: Contoh kode untuk menetapkan aturan untuk validasi data yang sudah dikumpulkan.

```
2 $rules = [
3     'username' => 'required|max_length[30]',
4     'password' => 'required|max_length[255]|min_length[10]',
5     'passconf' => 'required|max_length[255]|matches[password]',
6     'email'      => 'required|max_length[254]|valid_email',
7 ];
8 
```

10 Kode 2.64 merupakan contoh aturan yang ditetapkan untuk setiap *form* yang ada. Aturan-aturan
 11 tersebut dapat diganti sesuai dengan kebutuhan dari pengguna. Selain menggunakan aturan yang
 12 disediakan *CodeIgniter 4*, pengguna dapat membentuk aturannya sendiri pada file *Validation.php*.
 13 Kode 2.65 merupakan contoh aturan yang dibentuk secara manual.

Kode 2.65: Contoh kode pembentukan aturan secara manual pada file *Validation.php*.

```
14 <?php
15
16
17 class MyRules
18 {
19     public function even($value): bool
20     {
21         return (int) $value % 2 === 0;
22     }
23 }
24 
```

25 Kode 2.65 merupakan contoh pembentukan aturan secara manual. Penggunaan aturan yang
 26 dibentuk secara manual sama seperti penggunaan aturan lainnya.

27 2.3.8 *Helpers*

28 *Helpers* merupakan fungsi pada *CodeIgniter 4* yang menyediakan beberapa fungsi untuk pengguna
 29 dalam membangun aplikasi web. *Helpers* dapat dimuat oleh pengguna seperti berikut:

```
30 <?php
31     helper('helpers_name');
```

32 Setelah dilakukan pemanggilan, pengguna dapat memakai fungsi-fungsi yang disediakan sesuai
 33 dengan *helpers* yang digunakan. Fungsi-fungsi itu antara lain adalah *form*, *array*, dan lainnya.

34 2.4 Konversi CodeIgniter 3 ke CodeIgniter 4[3]

35 Konversi CodeIgniter 3 ke CodeIgniter 4 diperlukan penulisan ulang karena terdapat banyak
 36 implementasi yang berbeda. Konversi ke CodeIgniter 4 diawali dengan melakukan instalasi projek
 37 baru CodeIgniter 4. Instalasi dapat dilakukan dengan mengunduh file ataupun dapat dilakukan
 38 melalui *composer*.

39 2.4.1 Struktur Aplikasi

40 Struktur direktori pada CodeIgniter 4 memiliki perubahan yang terdiri *app*, *public*, dan *writable*.
 41 Direktori *app* merupakan perubahan dari direktori *application* dengan isi yang hampir sama dengan
 42 beberapa perubahan nama dan perpindahan direktori. Pada CodeIgniter 4 terdapat direktori *public*
 43 yang bertujuan sebagai direktori utama pada aplikasi website. Selanjutnya terdapat direktori
 44 *writable* yang berisikan *cache data*, *logs*, dan *session data*.

1 2.4.2 Routing

² CodeIgniter 4 meletakan *route* pada file app\Config\Routes.php. CodeIgniter 4 memiliki fitur
³ *auto routing* seperti pada CodeIgniter 3 namun, secara *default* di matikan. Fitur *auto routing*
⁴ memungkinkan untuk dinyalakan serupa dengan pada CodeIgniter 3 namun tidak direkomendasikan
⁵ karena alasan *security*.

6 2.4.3 Model, View, and Controller

7 Struktur MVC pada CodeIgniter 4 berbeda dibandingkan CodeIgniter 3 dimana terdapat perbedaan
8 penyimpanan direktori untuk ketiga *file* tersebut. Berikut merupakan penjelasan mengenai struktur
9 MVC:

10 *Model*

¹¹ Model terdapat pada direktori app\Models. Pembentukan file untuk Model perlu ditambahkan
¹² namespace App\Models; dan use CodeIgniter\Model; pada awal file setelah membuka tag PHP.
¹³ Selanjutnya nama fungsi perlu diubah dari extends CI_Model menjadi extends Model. Model
¹⁴ dapat dilakukan pembaharuan melalui cara berikut:

- 15 1. Pertama pengguna harus memindahkan seluruh *file model* menuju direktori app/Models
16 2. Selanjutnya pengguna harus menambahkan namespace App\Models; setelah pembukaan *tag*
17 *PHP*.
18 3. Pengguna juga harus menambahkan use CodeIgniter\Model; setelah kode diatas.
19 4. Pengguna harus mengganti extends CI_Model menjadi extends Model.
20 5. Terakhir pemanggilan *model* berubah dari sintaks:

menjadi sintaks berikut:

24 *View*

25 *View* pada CodeIgniter 4 terdapat di `app\Views` dengan sintaks yang harus diubah. Sintaks yang ha-
26 rus diubah merupakan sintaks untuk memanggil *view* pada CodeIgniter 3 `$this->load->view('x');`
27 sedangkan pada CodeIgniter 4 dapat menggunakan `return view('x');`. Selanjutnya, sintaks
28 `<?php echo $title?>` pada halaman *view* dapat diubah menjadi `<?= $title ?>`. Berikut meru-
29 pakan cara melakukan pembaharuan *view*:

- 30 1. Pertama pengguna perlu memindahkan seluruh *file views* menuju *app/Views*
31 2. Selanjutnya pengguna perlu mengubah sintaks:

33 menjadi sintaks berikut:

```
34         return view('directory_name/file_name');
```

- 35 3. Pengguna juga perlu mengubah sintaks:

1 \$content = \$this->load->view('file', \$data, TRUE);

2 menjadi sintaks berikut:

3 \$content = view('file', \$data);.

4 4. Pada *file views* pengguna dapat mengubah sintaks:

5 <?php echo \$title; ?>

6 menjadi sintaks berikut:

7 <?= \$title ?>.

8 5. Pengguna juga perlu menghapus apabila terdapat sintaks `defined('BASEPATH')` OR `exit('No`
`9 direct script access allowed');`.

10 *Controller*

11 *Controller* pada CodeIgniter 4 terdapat di `app\Controllers` dan diperlukan beberapa perubahan.

12 Pertama, perlu ditambahkan `namespace App\Controllers`; pada awal *file* setelah membuka tag

13 PHP. Selanjutnya, perlu mengubah `extends CI_Controller` menjadi `extends BaseController`.

14 Selanjutnya, diperlukan pengubahan nama pada pemanggilan *file* menjadi `App\Controllers\User.php`.

15 Pengguna dapat melakukan pembaharuan *controller* menggunakan cara berikut:

16 1. Pertama pengguna harus memindahkan seluruh *file controller* menuju `app\Controllers`.

17 2. Pengguna juga harus menambahkan sintaks `namespace App\Controllers`; setelah pembukaan *tag PHP*.

18 3. Selanjutnya pengguna harus mengubah `extends CI_Controller` menjadi `extends BaseController`.

19 4. Pengguna juga harus menghapus baris `defined('BASEPATH')` OR `exit('No direct script`
`20 access allowed');`; apabila ada.

22 2.4.4 *Class Loading*

23 Pada *CodeIgniter 4* sudah tidak terdapat *superobject* dengan komponen *framework* yang terinjeksi

24 sebagai properti pada *controller*. Kelas yang sudah dibentuk akan diinisiasi di tempat yang

25 membutuhkan dan komponen *framework* akan diatur oleh *service*. *Autoloader* pada *CodeIgniter 4*

26 secara otomatis menangani lokasi kelas dengan standar *PSR-4* di dalam direktori *App*.

27 2.4.5 *Configuration*

28 *File configuration* CodeIgniter 4 terdapat pada `app\Config` dengan penulisan sedikit berbeda dengan

29 versi sebelumnya. Penulisan berubah dari yang sebelumnya menggunakan *array* akan berubah

30 menjadi menggunakan variabel. Pengguna hanya perlu melakukan pemindahan menuju CodeIgniter

31 4 dan apabila menggunakan *file config custom* maka, diperlukan penulisan ulang pada direktori

32 *Config* dengan melakukan *extend* pada `CodeIgniter\Config\BaseConfig`. Beberapa konfigurasi

33 juga akan dipindahkan menuju file `.env`.

1 2.4.6 Database

2 Penggunaan *database* pada CodeIgniter 4 hanya berubah sedikit dibandingkan dengan versi se-
3 belumnya. Data-data penting kredensial diletakan pada `app\Config\Database.php` dan perlu
4 dilakukan beberapa perubahan sintaks dan *query*. Sintaks untuk memuat database diubah men-
5 jadi `$db = db_connect();` dan apabila menggunakan beberapa *database* maka sintaks menjadi
6 `$db = db_connect('group_name');`. Semua *query* harus diubah dari `$this->db` menjadi `$db`
7 dan beberapa sintaks perlu diubah seperti `$query->result()`; menjadi `$query->getResult()`;
8 Selain itu, terdapat *class* baru yakni *Query Builder Class* yang harus di inisiasi `$builder =`
9 `$db->table('mytable');` dan dapat dipakai untuk menjalankan *query* dengan mengganti `$this->db`
10 seperti `$this->db->get();` menjadi `$builder->get();`.

11 2.4.7 Migrations

12 Perubahan perlu dilakukan pada nama *file* menjadi nama dengan cap waktu. Selanjutnya dilakukan
13 penghapusan kode `defined('BASEPATH') OR exit('No direct script access allowed');`
14 dan menambahkan dua buah kode setelah membuka tag PHP yaitu:
15 • `namespace App\Database\Migrations;`
16 • `use CodeIgniter\Database\Migration;`
17 Setelah itu, `extends CI_Migration` diubah menjadi `extends Migration`. Terakhir, terdapat
18 perubahan pada nama metode *Forge* dari yang sebelumnya bernama `$this->dbforge->add_field`
19 menjadi menggunakan *camelCase* `$this->forge->addField`.

20 2.4.8 Routing

21 Pengguna dapat melakukan pembaharuan *routing* dengan cara berikut:
22 1. Pengguna dapat memakai *Auto Routing* [2.3.5](#) seperti pada *CodeIgniter 3* dengan menyalakan
23 *Auto Routing(Legacy)*.
24 2. Terdapat perubahan dari `(:any)` menjadi `(:segment)`.
25 3. Pengguna juga harus mengubah sintaks pada `app\Config\Routes.php` seperti berikut:
26 • `$route['journals'] = 'blogs';`
27 menjadi
28 `$routes->add('journals', 'Blogs::index');`
29 Sintak diatas berguna untuk memanggil fungsi `index` pada *controller Blogs*.
30 • `$route['product/(:any)'] = 'catalog/product_lookup'`
31 menjadi
32 `$routes->add('product/(:segment)', 'Catalog::productLookup');`

33 2.4.9 Libraries

34 CodeIgniter 4 menyediakan *library* untuk digunakan dan dapat diinstall apabila diperlukan. Pe-
35 manggilan *library* berubah dari `$this->load->library('x');` menjadi `$this->x = new X();`.
36 Terdapat beberapa *library* yang harus di perbaharui dengan sedikit perubahan. Berikut merupakan
37 beberapa *libraries* yang terdapat pembaharuan:

1 **Emails**

2 Perubahan *email* hanya terdapat pada nama dari *method* dan pemanggilan *library email*. Pemanggilan
3 *library* berubah dari `$this->load->library('email');` menjadi `$email = service('email');`;
4 dan selanjutnya perlu dilakukan perubahan pada semua `$this->email` menjadi `$email`. Selanjutnya
5 beberapa pemanggilan *method* berubah dengan tambahan *set* didepannya seperti *from* menjadi
6 *setFrom*.

7 **Working with Uploaded Files**

8 Terdapat banyak perubahan dimana pada CodeIgniter 4 pengguna dapat mengecek apakah *file* telah
9 terunggah tanpa *error* dan lebih mudah untuk melakukan penyimpanan *file*. Pada CodeIgniter 4
10 melakukan akses pada *uploaded file* dilakukan dengan sintaks berikut:

11 `$file = $this->request->getFile('userfile')`

12 selanjutnya dapat dilakukan validasi dengan cara sebagai berikut:

13 `$file->isValid()`

14 *File* yang sudah diunggah dapat disimpan dengan sintaks berikut:

15 `$path = $this->request->getFile('userfile')->store('head_img/', 'user_name.jpg');`

16 Sintaks diatas akan mengambil file dengan atribut nama `userfile` dan menyimpannya pada
17 direktori `head_img` dengan nama file `user_name.jpg`.

18 **HTML Tables**

19 Tidak terdapat banyak perubahan pada *framework* versi terbaru hanya perubahan pada nama *method*
20 dan pemanggilan *library*. Perubahan pemanggilan *library* dari `$this->load->library('table')`;
21 menjadi `$table = new \CodeIgniter\View\Table();` dan perlu dilakukan perubahan setiap
22 `$this->table` menjadi `$table`. Selain itu, terdapat beberapa perubahan pada penamaan *method*
23 dari *underscored* menjadi *camelCase*.

24 **Localization**

25 *CodeIgniter 4* mengembalikan *file* bahasa menjadi *array* sehingga perlu dilakukan beberapa perubahan.
26 Pertama, perlu dilakukan konfigurasi *default language* pada perangkat lunak. Selanjutnya melalui
27 pemindahan *file* bahasa pada *CodeIgniter 3* menuju `app\Language\<locale>`. *File-file* bahasa
28 *CodeIgniter 3* perlu dilakukan penghapusan semua kode `$this->lang->load($file, $lang);`
29 dan mengubah *method* pemanggilan bahasa dari `$this->lang->line('error_email_missing')`
30 menjadi `echo lang('Errors.errorEmailMissing');`

31 **Validations**

32 Pengguna dapat melakukan pembaharuan pada *validations* melalui cara berikut:

33 1. Pengguna harus mengubah kode pada *view* dari `<?php echo validation_errors(); ?>`
34 menjadi `<?= validation_list_errors() ?>`

1 2. Pengguna perlu mengubah beberapa kode pada *controller* seperti berikut:

- 2 • `$this->load->helper(array('form', 'url'));` menjadi `helper(['form', 'url']);`
- 3 • Pengguna perlu menghapus kode `$this->load->library('form_validation');`
- 4 • `if ($this->form_validation->run() == FALSE)` menjadi `if (!$this->validate([]))`
- 5 • `$this->load->view('myform');`
menjadi seperti berikut:

```
6                  return view('myform', ['validation' => $this->validator,]);
```

7 8 3. Pengguna juga perlu mengubah kode (dapat dilihat pada kode2.66) untuk melakukan validasi.

Kode 2.66: Perubahan kode untuk melakukan validasi.

```
9 10 <?php
11 12 $isValid = $this->validate([
13 14     'name' => 'required|min_length[3]',
15 16     'email' => 'required|valid_email',
17     'phone' => 'required|numeric|max_length[10]',
18 ]);
```

18 2.4.10 *Helpers*

19 *Helpers* tidak terdapat banyak perubahan namun, beberapa *helpers* pada *CodeIgniter 3* tidak
20 terdapat pada *CodeIgniter 4* sehingga perlu perubahan pada implementasi fungsinya. *Helpers* dapat
21 di dimuat secara otomatis menggunakan `app\Config\Autoload.php`

22 2.4.11 *Events*

23 *Events* merupakan pembaharuan dari *Hooks*. Pengguna harus mengubah

```
24                  $hook['post_controller_constructor']
```

25 menjadi

```
26                  Events::on('post_controller_constructor', ['MyClass', 'MyFunction']);}
```

27 Dan menambahkan `namespace CodeIgniter\Events\Events;.`

28 2.4.12 *Framework*

29 Pengguna tidak membutuhkan direktori *core* dan tidak membutuhkan kelas `MY_X` pada direktori
30 *libraries* untuk memperpanjang atau mengganti potongan *CI4*. Pengguna dapat membangun kelas
31 dimanapun dan menambahkan metode pada `app\Config\Services.php`.

1

BAB 3

2

ANALISIS

3 3.1 Analisis Sistem Kini

4 Seperti dibahas pada bab 2.2, *SharIF Judge* merupakan sebuah *online judge* yang dikustomisasi
5 sesuai dengan kebutuhan Informatika UNPAR. *SharIF Judge* dibentuk menggunakan *framework*
6 *CodeIgniter 3* yang menerapkan arsitektur *Model-View-Controller* atau *MVC*. Arsitektur ini me-
7 misahkan pemrosesan data pada *Model*, memisahkan logika pada *Controller*, dan memisahkan
8 tampilan pada *View*. Selain itu, terdapat direktori **assets** yang berisi seluruh kebutuhan
9 pengguna untuk ditampilkan seperti *javascript* dan gambar. Terakhir terdapat direktori **config** yang
10 berisi konfigurasi aplikasi dan *library* yang dibentuk dan digunakan oleh *SharIF Judge*.

11 **3.1.1 Model**

12 *Model* terdapat pada direktori **application/models**. Direktori ini berisi kelas *model* dengan
13 fungsi-fungsi untuk mengolah data pada aplikasi. Berikut merupakan *model* pada *SharIF Judge*
14 beserta fungsi-fungsinya.

15 **Assignment_model.php**

16 *Model Assignment* terdapat beberapa fungsi untuk memproses data pada tabel *assignment*. Berikut
17 merupakan fungsi-fungsi dari *model* tersebut:

- 18 • **add_assignment**
Fungsi ini berguna untuk menambahkan atau memperbarui *assignment* pada *database*.
- 19 • **delete_assignment**
Fungsi ini berguna untuk menghapus *assignment* pada *database*.
- 20 • **all_assignments**
Fungsi ini berguna untuk mengembalikan seluruh *assignment* beserta informasi *assignment*
21 tersebut.
- 22 • **new_assignment_id**
Fungsi ini berguna untuk mencari *id* terkecil yang dapat digunakan untuk menambahkan
23 *assignment* baru.
- 24 • **all_problems**
Fungsi ini berguna untuk mengembalikan seluruh *problems* dari *assignment* yang ada.
- 25 • **problem_info**
Fungsi ini berguna untuk mengembalikan baris tabel untuk *problem* tertentu.

- 1 • *assignment_info*
2 Fungsi ini berguna untuk mengembalikan baris tabel untuk **assignment** tertentu.
- 3 • *is_participant*
4 Fungsi ini berguna untuk mengecek apakah *username* merupakan peserta atau tidak.
- 5 • *increase_total_submits*
6 Fungsi ini berguna untuk menambahkan satu buah total *submit*.
- 7 • *set_moss_time*
8 Fungsi ini berguna untuk memperbarui "Moss Update Time" untuk *assignment* tertentu.
- 9 • *get_moss_time*
10 Fungsi ini berguna untuk mengembalikan "Moss Update Time" untuk *assignment* tertentu.
- 11 • *save_problem_description*
12 Fungsi ini berguna untuk menyimpan atau memperbarui deskripsi *problem*.
- 13 • *_update_coefficients*
14 Fungsi ini dipanggil pada fungsi *add_assignment* yang berguna untuk memperbarui koefisien
15 dari *assignment* tertentu.

16 **Hof_model.php**

17 Berikut merupakan fungsi-fungsi dari *Hof_model.php* yang berguna untuk mengambil data untuk
18 ditampilkan pada halaman *Hall of Fame*.

- 19 • *get_all_final_submission*
20 Fungsi ini berguna untuk mengambil data untuk *Hall of Fame* berdasarkan *username*.
- 21 • *get_all_user_assignments*
22 Fungsi ini berguna untuk mengambil detail *assignment* dan *problem* berdasarkan pengguna.

23 **Logs_model.php**

24 Berikut merupakan fungsi-fungsi dari *Logs_model.php* yang berguna untuk mencatat *log* pada
25 beberapa tabel.

- 26 • *insert_to_logs*
27 Fungsi ini berguna untuk mencatat *log* pada tabel *login*.
- 28 • *get_all_logs*
29 Fungsi ini berguna untuk mengembalikan seluruh *log* dalam bentuk *array*.

30 **Notifications_model.php**

31 *Notifications Assignment* terdapat beberapa fungsi untuk memproses data pada tabel *notifications*.

32 Berikut merupakan fungsi-fungsi dari *model* tersebut:

- 33 • *get_all_notifications*
34 Fungsi ini berguna untuk mengembalikan seluruh *notifications* dalam bentuk *array*.
- 35 • *get_latest_notifications*
36 Fungsi ini berguna untuk mengembalikan sepuluh *notification* terakhir.
- 37 • *add_notification*
38 Fungsi ini berguna untuk menambahkan *notification* baru.

- 1 • **update_notification**
2 Fungsi ini berguna untuk memperbarui *notification* tertentu.
- 3 • **delete_notification**
4 Fungsi ini berguna untuk menghapus *notification* tertentu.
- 5 • **get_notification**
6 Fungsi ini berguna untuk mengembalikan *notification* dalam bentuk *array*.
- 7 • **have_new_notification**
8 Fungsi ini berguna untuk mengecek apakah terdapat *notification* setelah waktu tertentu.

9 **Queue_model.php**

10 Berikut merupakan fungsi-fungsi dari *Queue_model.php* yang berguna untuk memproses data pada
11 halaman *queue*.

- 12 • **in_queue**
13 Fungsi ini berguna untuk mengecek apakah *submission* pengguna tertentu sudah dalam
14 antrean.
- 15 • **get_queue**
16 Fungsi ini berguna untuk mengembalikan data seluruh antrian.
- 17 • **empty_queue**
18 Fungsi ini berguna untuk menghapus seluruh tabel *queue*.
- 19 • **add_to_queue**
20 Fungsi ini berguna untuk memasukan *submission* kedalam tabel *queue*.
- 21 • **rejudge**
22 Fungsi ini berguna untuk menambahkan *submission* kedalam antrean untuk dilakukan *rejudge*.
- 23 • **rejudge_single**
24 Fungsi ini berguna untuk menambahkan satu buah *submission* kedalam antrean untuk dilaku-
25 kan *rejudge*.
- 26 • **get_first_item**
27 Fungsi ini berguna untuk mengambil data pertama dalam antrean.
- 28 • **remove_item**
29 Fungsi ini berguna untuk menghapus data tertentu dalam antrean.
- 30 • **save_judge_result_in_db**
31 Fungsi ini berguna untuk menyimpan hasil dari *judge* ke dalam *database*.
- 32 • **add_to_queue_exec**
33 Fungsi ini berguna untuk menambahkan data *dummy* pada antrean.

34 **Scoreboard_model.php**

35 Berikut merupakan fungsi-fungsi dari *Scoreboard_model.php* yang berguna untuk memproses data
36 untuk ditampilkan pada halaman *Score Board*.

- 37 • **_generate_scoreboard**
38 Fungsi ini dipanggil pada fungsi *update_scoreboard* dan berfungsi untuk menghasilkan
39 *scoreboard* dari *final submission*.

1 • **update_scoreboards**

2 Fungsi ini berguna untuk memperbarui *cache scoreboard* dari seluruh *assignment*. Fungsi
3 ini dipanggil setiap terdapat penghapusan pengguna atau seluruh *assignment* pengguna.

4 • **update_scoreboard**

5 Fungsi ini berguna untuk memperbarui *cache scoreboard* dari seluruh *assignment*. Fungsi
6 ini dipanggil setelah *judge* atau *rejudge*.

7 • **get_scoreboard**

8 Fungsi ini berguna untuk mengambil seluruh *cache scoreboard* dari *assignment* tertentu.

9 **Settings_model.php**

10 Berikut merupakan fungsi-fungsi dari **Settings_model.php** yang berguna untuk memproses data
11 untuk ditampilkan pada tabel *settings*.

12 • **get_setting**

13 Fungsi ini berguna untuk mengembalikan data *setting* tertentu.

14 • **set_setting**

15 Fungsi ini berguna untuk memperbarui *setting*.

16 • **get_all_settings**

17 Fungsi ini berguna untuk mengembalikan seluruh data *setting*.

18 • **set_settings**

19 Fungsi memperbarui beberapa *setting*.

20 **Submit_model.php**

21 Berikut merupakan fungsi-fungsi dari **Submit_model.php** yang berguna untuk memproses data
22 yang berkaitan dengan *submission*.

23 • **get_submission**

24 Fungsi ini berguna untuk mengembalikan baris data *submission* tertentu.

25 • **get_final_submissions**

26 Fungsi ini berguna untuk mengembalikan seluruh *final submission*.

27 • **get_all_submissions**

28 Fungsi ini berguna untuk mengembalikan seluruh *submission*.

29 • **count_final_submissions**

30 Fungsi ini berguna untuk menghitung seluruh *final submission*.

31 • **count_all_submissions**

32 Fungsi ini berguna untuk menghitung seluruh *submission*.

33 • **set_final_submission**

34 Fungsi ini berguna untuk memperbarui *submission* terentu menjadi *final submission*.

35 • **add_upload_only**

36 Fungsi ini berguna untuk menambahkan hasil dari *upload only* ke dalam *database*.

37 **User_model.php**

38 Berikut merupakan fungsi-fungsi dari **User_model.php** yang berguna untuk memproses data pada
39 tabel *users*.

- 1 • **have_user**
2 Fungsi ini berguna untuk mengecek apakah terdapat pengguna dengan *username* tertentu.
- 3 • **user_id_to_username**
4 Fungsi ini berguna untuk mengembalikan *user id* menjadi *username*.
- 5 • **username_to_user_id**
6 Fungsi ini berguna untuk mengembalikan *username* menjadi *user id*.
- 7 • **have_email**
8 Fungsi ini berguna untuk mengecek apakah terdapat *username* dengan *email* tertentu.
- 9 • **add_user**
10 Fungsi ini berguna untuk menambahkan sebuah pengguna.
- 11 • **add_users**
12 Fungsi ini berguna untuk menambahkan banyak pengguna.
- 13 • **delete_user**
14 Fungsi ini berguna untuk menghapus pengguna tertentu.
- 15 • **delete_submissions**
16 Fungsi ini berguna untuk menghapus seluruh *submission* pada pengguna tertentu.
- 17 • **validate_user**
18 Fungsi ini berguna untuk mengecek *username* dan *password* apakah sesuai dengan data.
- 19 • **selected_assignment**
20 Fungsi ini berguna untuk mengembalikan *assignment* untuk pengguna tertentu.
- 21 • **get_names**
22 Fungsi ini berguna untuk mengembalikan nama dari pengguna tertentu.
- 23 • **update_profile**
24 Fungsi ini berguna untuk memperbarui profil dari pengguna seperti *nama*, *email*, *password*,
25 dan *role*.
- 26 • **send_password_reset_mail**
27 Fungsi ini berguna untuk menghasilkan *password reset key* dan mengirim *email* untuk mela-
28 kukan *reset password*.
- 29 • **passchange_is_valid**
30 Fungsi ini berguna untuk mengecek apakah *password key* yang diberikan sesuai atau tidak.
- 31 • **reset_password**
32 Fungsi ini berguna untuk mengatur ulang *password* sesuai dengan *password key* tertentu.
- 33 • **get_all_users**
34 Fungsi ini berguna untuk mengembalikan seluruh pengguna untuk halaman *users*.
- 35 • **get_user**
36 Fungsi ini berguna untuk mengembalikan baris data untuk pengguna tertentu.
- 37 • **update_login_time**
38 Fungsi ini berguna untuk memperbarui *login time* dan *last login time* untuk pengguna
39 tertentu.

1 **User.php**

2 Berikut merupakan fungsi-fungsi dari `User.php` yang berguna untuk memproses data pada tabel
3 `users`.

- 4 • `select_assignment`

5 Fungsi ini berguna untuk mengatur *assignment* yang dipilih oleh pengguna.

- 6 • `save_widget_positions`

7 Fungsi ini berguna untuk memperbaharui posisi dari *dashboard widget* pada *database*.

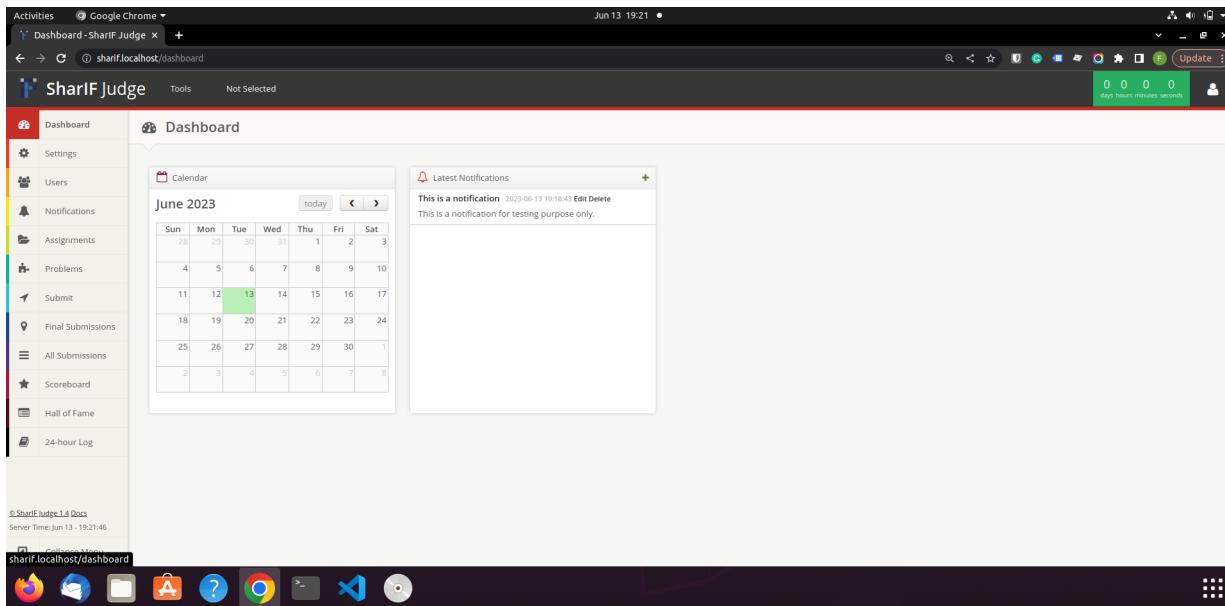
- 8 • `get_widget_positions`

9 Fungsi ini berguna untuk mengembalikan data *dashboard widget*.

10 **3.1.2 View**

11 *View* terdapat pada direktori `application/views`. Direktori ini berisikan seluruh *file* untuk tampilan halaman *SharIF Judge*. *File* tersebut dipisahkan oleh direktori sesuai dengan fungsinya.
12 Direktori tersebut dibagi menjadi tiga buah direktori utama yakni *error*, *pages*, dan *templates*.
13 Direktori *error* berisikan tampilan halaman *error* yang akan dilihat oleh pengguna. Direktori *pages* merupakan tampilan utama *SharIF Judge* yang terbagi lagi menjadi dua buah direktori yakni *admin* dan *authentication*. Direktori *admin* berisikan tampilan halaman untuk *role admin*. Direktori *authentication* berisikan tampilan halaman untuk akses pengguna seperti *Login*, *Register*, dan *Reset Password*. Direktori *templates* terdiri dari tampilan yang digunakan oleh seluruh tampilan halaman seperti *header* dan *side bar*. Berikut merupakan tampilan halaman pada *SharIF Judge*:

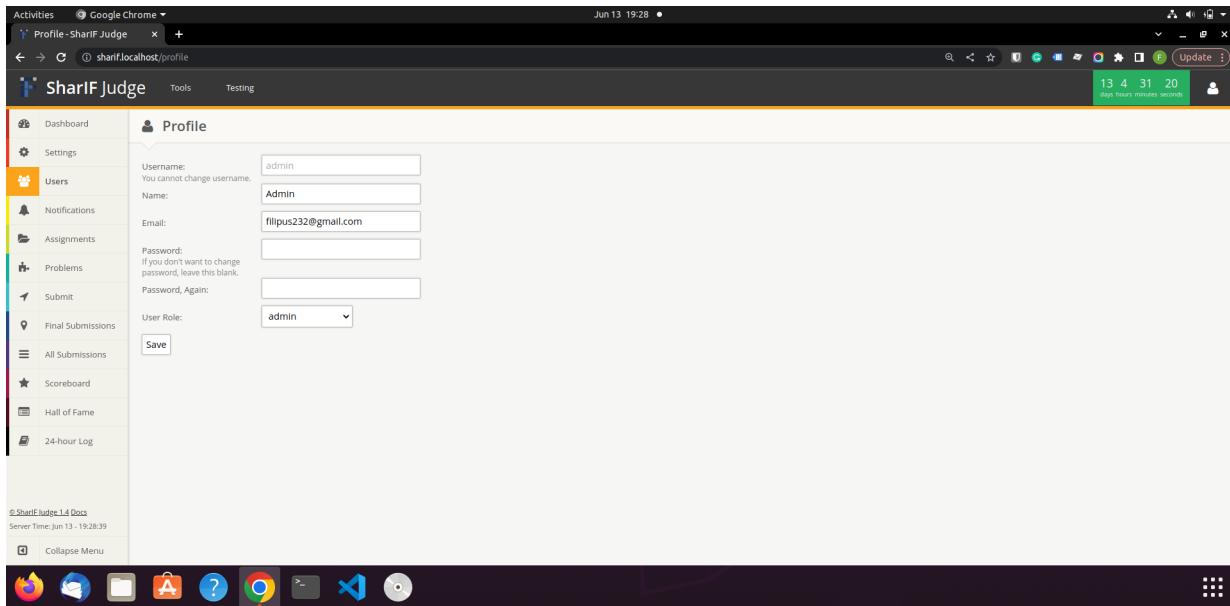
20 **Dashboard**



Gambar 3.1: Tampilan Halaman Dashboard

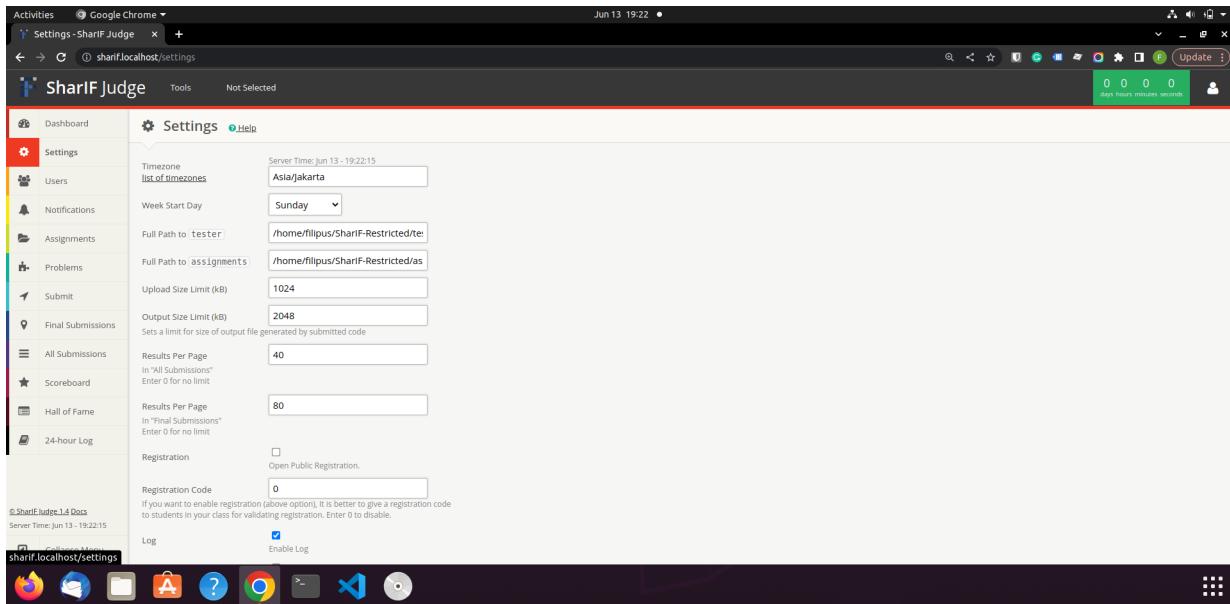
21 Gambar 3.1 merupakan tampilan halaman *dashboard* yang terdapat pada semua *role* pengguna.

1 Profile



Gambar 3.2: Tampilan Halaman Profile

- 2 Gambar 3.2 merupakan tampilan halaman *profile* yang terdapat pada semua *role* pengguna. Namun, terdapat fitur yang tidak dapat digunakan oleh *siswa* dan *instructor* yakni mengganti role.
- 4 *Settings*



Gambar 3.3: Tampilan Halaman Settings

- 5 Gambar 3.3 merupakan tampilan halaman *settings* yang terdapat hanya pada *role* admin dan *head instructor*.

1 *Users*

#	User ID	Username	Display Name	Email	Role	First Login	Last Login	Actions
1	1	admin	Admin	filipus232@gmail.com	admin	2022-12-15 17:17:42	2023-06-13 19:17:53	
2	2	instructor	Instructor One	instructor@sharifjudge.ir	head_instructor	2022-12-16 12:19:44	2022-12-16 12:19:44	
3	4	i19074	Student One	st1@sharifjudge.ir	student	2022-12-16 12:13:55	2022-12-20 01:00:04	
4	5	student2	Student Two	st2@sharifjudge.ir	student	Never	Never	
5	6	student3	Student Three	st3@sharifjudge.ir	student	Never	Never	
6	7	student4	Student Four	st4@sharifjudge.ir	student	Never	Never	
7	8	student5	Student Five	st5@sharifjudge.ir	student	Never	Never	
8	9	student6	Student Six	st6@sharifjudge.ir	student	Never	Never	

Gambar 3.4: Tampilan Halaman Users

- 2 Gambar 3.4 merupakan tampilan halaman *users* yang terdapat hanya pada *role* admin dan *head instructor*.

4 *Notifications*

This is a notification
2023-06-13 19:18:43 [Edit](#) [Delete](#)

This is a notification for testing purpose only.

Gambar 3.5: Tampilan Halaman Notifications

- 5 Gambar 3.5 merupakan tampilan halaman *notifications* yang terdapat pada semua *role* pengguna.

1 *Assignments*

The screenshot shows the Sharf Judge web application interface. The title bar indicates it's running in Google Chrome on June 13 at 19:23. The main content area is titled 'Assignments' and shows a single assignment entry:

Select	Name	Problems	Submissions	Coefficient	Start Time	Finish Time	Status	PDF	Actions
<input type="checkbox"/>	Testing	1 problem	0 submissions	100 %	2023-06-13 00:00:00	2023-06-27 00:00:00	Close		

Below the table, there are two notices: 'Assignment added successfully.' and 'Notice: You did not upload any zip file for tests. If needed, upload by editing assignment.' and 'Notice: You did not upload any pdf file for assignment. If needed, upload by editing assignment.'

Gambar 3.6: Tampilan Halaman Assignments

- 2 Gambar 3.6 merupakan tampilan halaman *assignments* yang terdapat pada semua *role* pengguna.
- 3 Namun, terdapat bagian yang tidak dapat diakses oleh *role* siswa dan *instructor* yakni bagian *actions*.

5 *Problems*

The screenshot shows the Sharf Judge web application interface. The title bar indicates it's running in Google Chrome on June 13 at 19:24. The main content area is titled 'Problem' and shows a message 'Description not found'. On the right side, there are two panels: 'Testing' and 'Submit'. The 'Testing' panel shows a table with one problem entry:

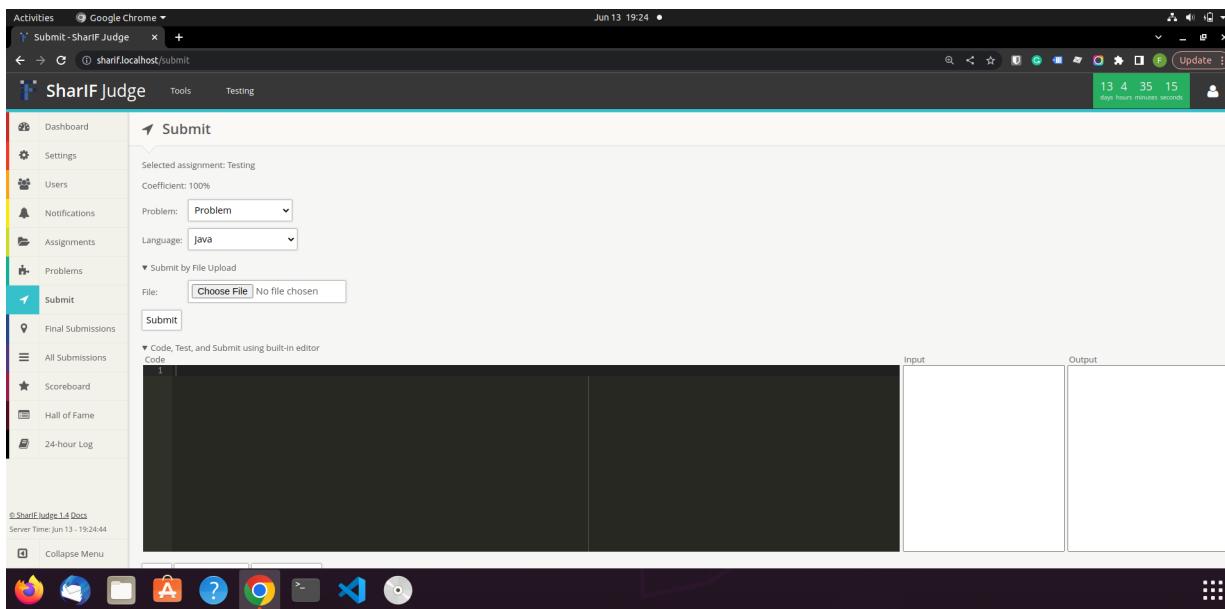
#	Problem	Score	Upload Only
1	Problem	100	No

The 'Submit' panel has a dropdown menu for 'Select Language' which is currently open, showing options: 'C', 'C++', 'Python 2', 'Python 3', and 'Java'.

Gambar 3.7: Tampilan Halaman Problems

- 6 Gambar 3.7 merupakan tampilan halaman *problems* yang terdapat pada semua *role* pengguna.

1 Submit



Gambar 3.8: Tampilan Halaman Submit

2 Gambar 3.8 merupakan tampilan halaman *submit* yang terdapat pada semua *role* pengguna.

3 Final Submissions

#	ID	Username	Name	Problem	Submit Time	Score	Delay %	Final Score	Language	Status	Code	Log	Actions
1	1	admin	Admin	1	2023-06-13 19:26:24	0	No Delay 100%	0	java	Compilation Error	Code	Log	

Gambar 3.9: Tampilan Halaman Final Submission

4 Gambar 3.9 merupakan tampilan halaman *submit* yang terdapat pada semua *role* pengguna.

1 **All Submissions**

Final	ID	Username	Name	Problem	Submit Time	Score			Language	Status	Code	Log	Actions
						Score	Delay %	Final Score					
●	2	admin	Admin	1	2023-06-13 19:28:03	0	No Delay 100%	0	Py 3	Syntax Error	Code	Log	✖
○	1	admin	Admin	1	2023-06-13 19:26:24	0	No Delay 100%	0	Java	Compilation Error	Code	Log	✖

Gambar 3.10: Tampilan Halaman All Submission

- 2 Gambar 3.10 merupakan tampilan halaman *All Submission* yang terdapat pada semua *role* pengguna.

3 **Scoreboard**

#	Username	Name	Problem	Total
1	admin	Admin	100 19:26	100 19:31

Gambar 3.11: Tampilan Halaman Scoreboard

- 4 Gambar 3.10 merupakan tampilan halaman *All Submission* yang terdapat pada semua *role* pengguna.

1 **Hall of Fame**

Rank	Username	Display Name	Total Score
1	admin	Admin	0

Gambar 3.12: Tampilan Halaman Hall of Fame

2 Gambar 3.12 merupakan tampilan halaman *Hall of Fame* yang terdapat pada semua *role* pengguna.

3 **24-hour Log**

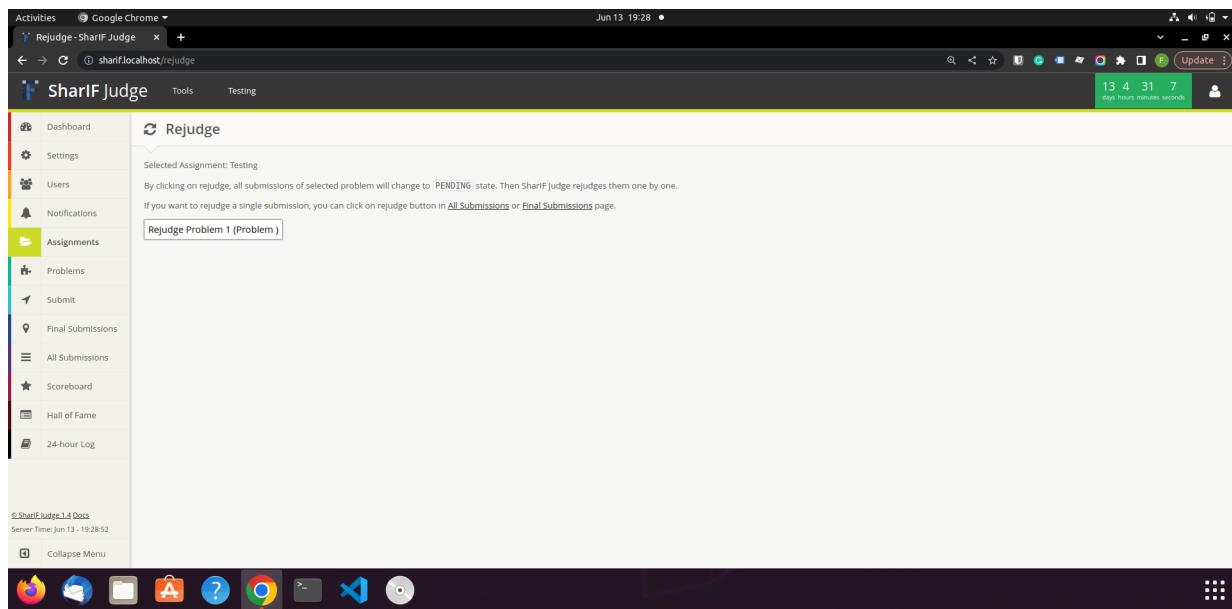
The table shows one row of data:

#	Login ID	Username	IP Address	Login Time	Log from different IP (< 24 hours)
1	admin	admin	127.0.0.1	Jun 13 19:28:16	

Gambar 3.13: Tampilan Halaman 24-hour Log

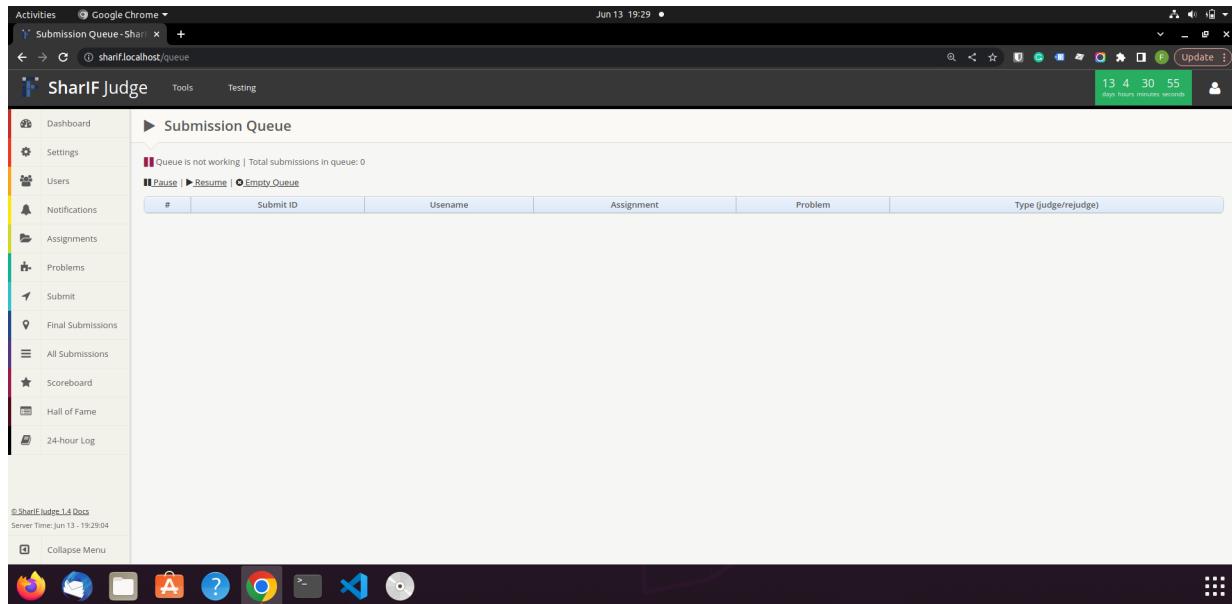
4 Gambar 3.13 merupakan tampilan halaman *24-hour Log* yang terdapat hanya pada *role admin* dan *head instructor*.

1 Rejudge



Gambar 3.14: Tampilan Halaman ReJudge

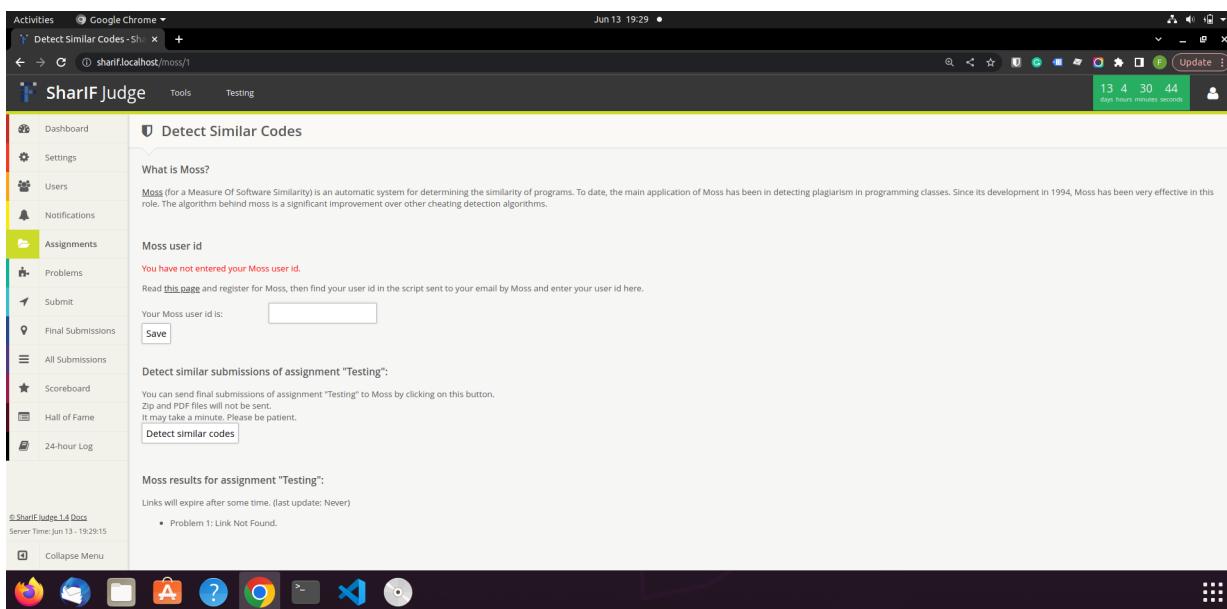
- 2 Gambar 3.14 merupakan tampilan halaman *ReJudge* yang terdapat hanya pada *role admin* dan *head instructor*.
- 4 ***Submission Queue***



Gambar 3.15: Tampilan Halaman Submission Queue

- 5 Gambar 3.15 merupakan tampilan halaman *Submission Queue* yang terdapat hanya pada *role admin* dan *head instructor*.

1 Cheat Detection



Gambar 3.16: Tampilan Halaman Cheat Detection

- 2 Gambar 3.16 merupakan tampilan halaman *Cheat Detection* yang terdapat hanya pada *role admin*
3 dan *head instructor*.

4 3.1.3 Controller

- 5 *Controller* pada direktori `application/controller`. Direktori ini berisikan kelas *controller* dengan
6 fungsi-fungsi dalam mengambil atau memberikan data *models* untuk dialihkan menuju *views* untuk
7 ditampilkan. Berikut merupakan *controller* pada *SharIF Jugde* beserta fungsi-fungsinya.

8 Assignments.php

- 9 Berikut merupakan fungsi-fungsi pada *controller* `Assignments.php`.

- 10 • **index**
Fungsi ini berguna untuk mengambil dan memberikan data menuju halaman `assignments.twig`
11 menggunakan `Assignment_model`.
- 13 • **select**
Fungsi ini berguna untuk memilih *assignment* menggunakan *ajax*.
- 15 • **pdf**
Fungsi ini berguna untuk mengunduh *assignment* atau *problem* dalam bentuk pdf.
- 17 • **downloadtestsdesc**
Fungsi ini berguna untuk mengunduh dan mengompres data *test* dan deskripsi sebuah
18 *assignment*.
- 20 • **download_submissions**
Fungsi ini berguna untuk mengunduh dan mengompres kode terakhir sebuah *assignment*
22 pengguna.

- 1 • **delete**
- 2 Fungsi ini berguna untuk menghapus *assignment*.
- 3 • **add**
- 4 Fungsi ini berguna untuk menambah atau mengubah *assignment* berdasarkan masukan pengguna.
- 5 • **_add**
- 6 Fungsi ini berguna untuk menambah atau mengubah *assignment*.
- 7 • **edit**
- 8 Fungsi ini berguna untuk mengecek *role* pengguna dapat mengubah *assignment*. Selanjutnya akan dikembalikan pada fungsi **add**.
- 9 • **pdfCheck** Fungsi ini berguna untuk mengecek *file pdf* dari sebuah *assignment*.

12 **Dashboard.php**

13 Berikut merupakan fungsi-fungsi pada *controller Dashboard.php*.

- 14 • **index**
- 15 Fungsi ini berguna untuk mengambil dan memberikan data menuju halaman *dashboard* menggunakan tiga buah *model*. *Model* tersebut terdiri dari **Assignment_model**, **Settings_model**, dan **Notifications_model**.
- 16 • **widget_positions**
- 17 Fungsi ini berguna untuk menyimpan data *widget* pengguna.

20 **Install.php**

21 *Controller Install.php* hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna untuk membentuk tabel yang dibutuhkan oleh *SharIF Judge* pada *database*. Selain itu, fungsi ini juga berguna untuk memasukan data pengguna *admin* yang pertama kali memasang *SharIF Judge* pada perangkat.

25 **Login.php**

26 Berikut merupakan fungsi-fungsi pada *controller Login.php*.

- 27 • **_registration_code**
- 28 Fungsi ini berguna untuk memeriksa kode registrasi.
- 29 • **index**
- 30 Fungsi ini berguna untuk melakukan validasi *username* dan *password* pengguna. Selain itu, fungsi ini juga memperbarui *log* pada tabel *login*.
- 31 • **register**
- 32 Fungsi ini berguna untuk melakukan validasi dalam pembetukan akun.
- 33 • **logout**
- 34 Fungsi ini berguna untuk menghancurkan *session* dari pengguna dan memindahkan pengguna ke halaman *login*.
- 35 • **lost**
- 36 Fungsi ini berguna untuk mengirim *email lupa password*.

- 1 • **rest**

2 Fungsi ini berguna untuk melakukan *reset password* pengguna.

3 **Logs.php**

4 *Controller Logs.php* hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna untuk
5 mengambil dan memberikan data pada halaman *logs* menggunakan **Logs_model**.

6 **Moss.php**

7 Berikut merupakan fungsi-fungsi pada *controller Moss.php*.

- 8 • **index**

9 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *moss*.

- 10 • **update**

11 Fungsi ini berguna untuk memperbarui *moss_userid* yang dimasukan oleh pengguna.

- 12 • **_detec**

13 Fungsi ini berguna untuk melakukan pengecekan terhadap *submission*. !!TODO

14 **Notification.php**

15 Berikut merupakan fungsi-fungsi pada *controller Notification.php*.

- 16 • **index**

17 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *notifications*
18 menggunakan **assignment_model** dan **notifications_model**.

- 19 • **add**

20 Fungsi ini berguna untuk menambahkan data *notifications*.

- 21 • **edit**

22 Fungsi ini berguna untuk memperbarui data *notifications*.

- 23 • **delete**

24 Fungsi ini berguna untuk menghapus data *notifications*.

- 25 • **check**

26 Fungsi ini berguna memeriksa *notifications* baru.

27 **Problems.php**

28 Berikut merupakan fungsi-fungsi pada *controller Problems.php*.

- 29 • **index**

30 Fungsi ini berguna untuk mengambil dan memberikan data *problems* sesuai dengan *assignment*
31 tertentu pada halaman *problems*.

- 32 • **edit**

33 Fungsi ini berguna untuk memperbarui deskripsi *problems* pada *assignment* tertentu.

34 **Profile.php**

35 Berikut merupakan fungsi-fungsi pada *controller Profile.php*.

1 • **index**

2 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *profile*. Selain itu,
3 fungsi ini berguna untuk melakukan pembaharuan data *profile*.

4 • **_password_check**

5 Fungsi ini berguna untuk melakukan validasi terhadap *password* yang akan dimasukkan
6 pengguna sesuai dengan aturan.

7 • **_password_again_check**

8 Fungsi ini berguna untuk melakukan validasi terhadap pengulangan *password* yang dimasukkan
9 pengguna.

10 • **_email_check**

11 Fungsi ini berguna untuk melakukan validasi terhadap *email* yang dimasukkan pengguna

12 • **_role_check**

13 Fungsi ini berguna untuk melakukan validasi *role* pengguna.

14 **Queue.php**

15 • **index**

16 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *queue* menggunakan
17 tiga buah *model*. *Model* tersebut adalah **Assignment_model**, **queue_model**, dan
18 **settings_model**.

19 • **pause**

20 Fungsi ini berguna untuk memperbarui data pada tabel *settings*.

21 • **resume**

22 Fungsi ini berguna untuk melanjutkan proses *queue*.

23 • **empty_queue**

24 Fungsi ini berguna untuk menghapus data tabel *queue*.

25 **Queueprocess.php**

26 **Controller Queueprocess.php** hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna
27 untuk menjalankan proses *judge* berdasarkan *queue* satu demi satu sesuai antrean. Fungsi
28 ini menggunakan beberapa *model* yaitu **Queue_model**, **Submit_model**, **Assignments_model**, dan
29 **Settings_model**.

30 **Rejudge.php**

31 Berikut merupakan fungsi-fungsi pada *controller Rejudge.php*.

32 • **index**

33 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *rejudge* menggunakan
34 *Assignment_model*.

35 • **rejudge_single**

36 Fungsi ini berguna untuk melakukan *rejudge* pada satu buah masalah tertentu.

1 **Scoreboard.php**

2 Berikut merupakan fungsi-fungsi pada *controller Scoreboard.php*.

- 3 • **index**

4 Fungsi ini berguna untuk memberikan data pada halaman *scoreboard* menggunakan dua buah
5 *model*. *Model* tersebut adalah **Assignment_model** dan **Scoreboard_model**.

6 **Server_time.php**

7 *Controller Server_time.php* hanya memiliki satu buah fungsi bernama **index**. Fungsi ini berguna
8 untuk mengeluarkan *server_time*.

9 **Settings.php**

10 Berikut merupakan fungsi-fungsi pada *controller Settings.php*.

- 11 • **index**

12 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *settings* menggunakan
13 *Settings_model* dan **Assignment_model**.

- 14 • **update**

15 Fungsi ini berguna untuk mengambil masukan dan memperbarui data pada halaman
16 *settings* berdasarkan masukan tersebut. Data tersebut nantinya akan disimpan pada *database*
17 menggunakan fungsi **Settings_model**.

18 **Submission.php**

19 Berikut merupakan fungsi-fungsi pada *controller Submission.php*.

- 20 • **_download_excel**

21 Fungsi ini berguna untuk mengubah data-data dari *submission* yang dipilih menjadi format
22 *excel*.

- 23 • **final_excel**

24 Fungsi ini berguna untuk mengunduh data *final submissions*.

- 25 • **all_excel**

26 Fungsi ini berguna untuk mengunduh data seluruh *submissions*.

- 27 • **the_final**

28 Fungsi ini berguna untuk memberikan data pada halaman *Final Submissions* menggunakan beberapa
29 *model*. *Model* tersebut terdiri dari **Submit_model**, **Settings_model**, dan **User_model**.

- 30 • **all**

31 Fungsi ini berguna untuk memberikan data pada halaman *All Submissions* menggunakan beberapa
32 *model*. *Model* tersebut terdiri dari **Submit_model**, **Settings_model**, dan **User_model**.

- 33 • **select**

34 Fungsi ini berguna untuk memilih *submission* yang akan dijadikan *submission final* oleh
35 pengguna.

- 36 • **_check_type**

37 Fungsi ini berguna untuk melakukan pengecekan tipe *submission* yang telah dikumpulkan
38 oleh pengguna.

1 • **view_code**

2 Fungsi ini berguna untuk memperlihatkan *submission* yang telah dikumpulkan oleh pengguna
3 sesuai dengan tipenya.

4 • **download_file**

5 Fungsi ini berguna untuk mengunduh hasil dari *submission* yang telah dikumpulkan oleh
6 pengguna.

7 **Submit.php**

8 Berikut merupakan fungsi-fungsi pada *controller* **Submit.php**.

9 • **_language_to_type**

10 Fungsi ini berguna untuk mengubah bahasa pemrograman menjadi tipe sesuai dengan pilihan
11 pengguna.

12 • **_language_to_ext**

13 Fungsi ini berguna untuk mengubah bahasa pemrograman menjadi ekstensi sesuai dengan
14 pilihan pengguna.

15 • **_match**

16 Fungsi ini berguna untuk mencocokan tipe dengan ekstensi dari bahasa pemrogramannya.

17 • **_check_language**

18 Fungsi ini berguna untuk melakukan pengecekan terhadap bahasa pemrograman yang digu-
19 nakan.

20 • **index**

21 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *submit* menggunakan
22 **Assignment_model**.

23 • **_upload**

24 Fungsi ini berguna untuk menyimpan jawaban dan memasukannya ke *queue* untuk dinilai.

25 • **load**

26 Fungsi ini berguna untuk memuat kode dari *editor file*.

27 • **save**

28 Fungsi ini berguna untuk meyimpan kode menuju *editor file* dan mengirim ataupun menjalan-
29 kannya.

30 • **_submit**

31 Fungsi ini berguna untuk menambahkan kode pada *queue* untuk dilakukan *judge*.

32 • **_execute**

33 Fungsi ini berguna untuk menambahkan kode untuk dijalankan atau di *queue*.

34 • **get_output** Fungsi ini berguna untuk memuat *file* menjadi hasil eksekusi.

35 **User.php**

36 Berikut merupakan fungsi-fungsi pada *controller* **User.php**.

37 • **index**

38 Fungsi ini berguna untuk mengambil dan memberikan data pada halaman *users*.

39 • **add**

40 Fungsi ini berguna untuk menambahkan pengguna baru sesuai dengan masukan.

- 1 • **delete**
2 Fungsi ini berguna untuk menghapus pengguna yang dipilih.
- 3 • **delete_submissions**
4 Fungsi ini berguna untuk menghapus *submission* dari sebuah pengguna.
- 5 • **list_excel**
6 Fungsi ini berguna untuk menghasilkan dan mengunduh data pengguna pada format *excel*.

7 **3.1.4 Assets**

8 Direktori ini berisikan seluruh kebutuhan pengguna seperti *library javascript* dan juga gambar yang
9 akan ditampilkan pada aplikasi. Berikut merupakan isi dari direktori ini beserta kegunaannya.

- 10 • Direktori *ace*
11 Direktori ini berisikan *javascript* yang berfungsi untuk menambahkan *code editor* pada aplikasi.
- 12 • Direktori *font*
13 Direktori ini berisikan seluruh *font* yang digunakan oleh aplikasi *SharIF Judge*.
- 14 • Direktori *fullcalendar*
15 Direktori ini berisikan *javascript* dan *css* yang berfungsi untuk menambahkan kalender pada
16 aplikasi.
- 17 • Direktori *gridster*
18 Direktori ini berisikan *javascript*, *css*, dan gambar yang berfungsi untuk membentuk kolom
19 *grid* dengan sifat *drag and drop*.
- 20 • Direktori *images*
21 Direktori ini berisikan seluruh gambar yang digunakan pada aplikasi *SharIF Judge* seperti
22 banner dan juga logo.
- 23 • Direktori *js*
24 Direktori ini berisikan *javascript* dan *jquery*. *Javascript* dan *jquery* yang terdapat pada
25 direktori ini dibentuk secara manual dan diambil dari situs resmi.
- 26 • Direktori *nano_scroller*
27 Direktori ini berisikan *jquery* dan *css* yang berfungsi untuk membentuk *scrollbar* pada aplikasi.
- 28 • Direktori *noty*
29 Direktori ini berisikan *javascript* yang berfungsi untuk mempermudah membentuk pesan *alert*,
30 *success*, dan sebagainya.
- 31 • Direktori *pdfjs*
32 Direktori ini berisikan *javascript* yang berfungsi untuk melakukan *parsing* dan *rendering* file
33 *pdf*.
- 34 • Direktori *reveal*
35 Direktori ini berisikan *jquery* dan *css* yang berfungsi untuk menampilkan *popup* berupa
36 halaman yang dikonfigurasikan.
- 37 • Direktori *snippet*
38 Direktori ini berisikan *javascript* dan *css* yang berfungsi sebagai *template* dalam membentuk
39 kode *javascript*.
- 40 • Direktori *styles*
41 Direktori ini berisikan *css* yang berfungsi untuk memperindah aplikasi yang dibentuk.

- 1 • Direktori *tinymce*

2 Direktori ini berisikan *javascript* dan *css* yang berfungsi untuk membentuk *WYSIWYG editor*
3 pada sebuah aplikasi.

4 **3.1.5 Config**

5 Direktori ini berisikan seluruh konfigurasi aplikasi *SharIF Judge* seperti *database*, *url* aplikasi, dan
6 lainnya. Berikut merupakan isi dari direktori ini beserta kegunaannya.

- 7 • **autoload.php**

8 File ini berisikan seluruh data yang ingin diinisiasi saat aplikasi dijalankan seperti *libraries*,
9 *helpers*, dan *model*.

- 10 • **config.php**

11 File ini berisikan konfigurasi aplikasi *SharIF Judge* seperti *base url* aplikasi, *encryption key*,
12 konfigurasi *session*, konfigurasi *cookie*, dan *csfr*.

- 13 • **constants.php**

14 File ini berisikan data yang bersifat *global* pada aplikasi.

- 15 • **database.php**

16 File ini berisikan konfigurasi *database* yang digunakan oleh pengguna.

- 17 • **doctypes.php**

18 File ini berisikan deklarasi tipe dokumen untuk *HTML*.

- 19 • **foreign_chars.php**

20 File ini terjemahan data *foreign character* yang terdapat pada *text helper*.

- 21 • **memcached.php**

22 File ini berisikan konfigurasi *server* untuk menyimpan data *cache* pada server.

- 23 • **migration.php**

24 File ini berisikan konfigurasi untuk melakukan *migration* seperti tipe *migration* dan *path file migration* disimpan.

- 26 • **mime.php**

27 File ini berisikan tipe-tipe *mime* yang digunakan oleh *upload helper*.

- 28 • **profiler.php**

29 File ini berisikan konfigurasi untuk *profiler*.

- 30 • **routes.php**

31 File ini berisikan konfigurasi *route* dan *route* yang didefinisikan secara manual.

- 32 • **secrets.php**

33 File ini dibentuk oleh pengguna yang berisikan data-data untuk melakukan autentikasi dan
34 konfigurasi *email*.

- 35 • **smiley.php**

36 File ini berisikan terjemahan dari sintaks *emoticon* menjadi gambar yang digunakan pada
37 *emoticon helper*.

- 38 • **twig.php**

39 File ini berisikan konfigurasi *template engine twig*.

- 40 • **user_agents.php**

41 File ini berisikan tipe dari *platform*, *browser*, dan lainnya.

3.1.6 Libraries

SharIF Judge menggunakan beberapa *library* yang dibentuk secara manual maupun yang sudah tersedia pada *Codeigniter 3*. Berikut merupakan *library* yang dipakai oleh *SharIF Judge*:

Unzip

Unzip merupakan sebuah *library* yang dibentuk oleh Phil Sturgeon. *Library* ini mewajibkan pengguna untuk menyalakan *extension Zlib* sebelum dapat digunakan. *Library Unzip* berfungsi untuk mengextraksi file dengan *extension .zip* menuju direktori yang ditentukan dan dapat mengeluarkan *error* yang sesuai. *Library* ini juga dapat memberi batasan *extension* apa yang diinginkan dari file tersebut. Kode 3.1 merupakan contoh penggunaan *library Unzip* pada *SharIF Judge*.

Kode 3.1: Contoh kode penggunaan *Library Unzip*

```
11 $this->load->library('unzip');
12 $this->unzip->allow(array('txt', 'cpp', 'html', 'md', 'pdf'));
13 $extract_result = $this->unzip->extract($u_data['full_path'], $tmp_dir);
```

Kode 3.1 merupakan contoh penggunaan *library Unzip*. Sintaks `$this->load->library('unzip');` berfungsi untuk melakukan *load library Unzip* agar dapat digunakan pada fungsi tersebut. Selanjutnya sintaks `$this->unzip->allow(array('txt', 'cpp', 'html', 'md', 'pdf'));` berfungsi untuk memberikan batasan *extension* apa saja yang diinginkan dari file tersebut. Terakhir sintaks `$extract_result = $this->unzip->extract($u_data['full_path'], $tmp_dir);` berfungsi untuk melakukan ekstraksi terhadap file zip pada `$u_data['full_path']` tersebut menuju direktori pada variabel `tmp_dir`.

Twig

Twig merupakan sebuah *template engine library* yang digunakan untuk mempermudah dalam membentuk *view* pada aplikasi. *Twig* terintegrasi dengan fungsi-fungsi pada *CodeIgniter 3* sehingga dapat menggunakan seluruh fungsi yang terdapat pada *CodeIgniter 3*. Kode 3.2 merupakan contoh penggunaan *Twig* pada *SharIF Judge*.

Kode 3.2: Contoh *view* menggunakan *library Twig*

```
28 29 1  {% set title = 'Login' %} 30 2  {% include 'templates/simple_header.twig' %} 31 3 32 4  {{ form_open() }} 33 5  <div class="box login"> 34 6 35 7  <div class="judge_logo"> 36 8  <a href="{{ site_url() }}></a> 37 9  </div> 38 0 39 1  <div class="login_form"> 40 2  <div class="login1"> 41 3  <p> 42 4  <label for="form_username">Username</label><br/> 43 5  <input id="form_username" type="text" name="username" required="required" pattern="[0-9a-zA-Z]{3,20}" title="The 44  Username field must be between 3 and 20 characters in length, and contain only digits and lowercase 45  letters" class="sharif_input" value="{{ set_value('username') }}" autofocus="autofocus"/> 46 6  {{ form_error('username', '<div class="shj_error">', '</div>') }} 47 7  </p> 48 8  <p> 49  <label for="form_password">Password</label><br/> 50 0  <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The 51  Password field must be at least 6 characters in length" class="sharif_input"/>
```

```

21         {{ form_error('password', '<div class="shj_error">', '</div>') }}
22     </p>
23     {% if error %}
24         <div class="shj_error">Incorrect username or password.</div>
25     {% endif %}
26     </div>
27     <div class="login2">
28         <p style="margin:0;">
29             {% if registration_enabled %}
30                 <a href="{{ site_url('register') }}>Register</a> |
31             {% endif %}
32                 <a href="{{ site_url('login/lost') }}>Reset Password</a>
33             <input type="submit" value="Login" id="sharif_submit"/>
34         </p>
35     </div>
36     </div>
37
38     </div>
39 </form>
40 </body>
41 </html>

```

Twig pada *view SharIF Judge* menggunakan dua buah *delimiters* yakni `{}` dan `{% %}`. *Delimiters* `{}` memiliki fungsi untuk mengembalikan *expression* seperti variabel ataupun fungsi *CodeIgniter 3*. Contoh fungsi yang dikembalikan oleh *delimiters* pada kode diatas adalah `form_open` yang merupakan sebuah fungsi pada *CodeIgniter 3* untuk membuka *tag form*. Sedangkan *delimiters* `{% %}` memiliki fungsi untuk mengeksekusi fungsi PHP seperti *for-loops* atau *if else*. Contoh fungsi yang dieksekusi pada kode diatas adalah `if` yang berfungsi untuk mengecek kondisi tertentu.

29 ***Password_hash***

30 *Password_hash* merupakan sebuah *library* yang dibentuk oleh *phpass*. *Library* ini berfungsi untuk
31 melakukan enkripsi *password* dan melakukan verifikasi *password*. *Library* ini mendukung beberapa
32 metode enkripsi antara lain *CRYPT_BLOWFISH* dan *CRYPT_EXT_DES*. Kode 3.3 merupakan
33 contoh penggunaan *library* ini pada *SharIF Judge*.

Kode 3.3: Contoh kode penggunaan *Library Password_hash*

```

34
35 1   $this->load->library('password_hash', array(8, FALSE));
36 2   $user['password'] = $this->password_hash->HashPassword($this->input->post('password'));

```

38 Kode 3.3 merupakan contoh penggunaan *library Passwors_hash*. Sintaks pada baris pertama
39 akan melakukan *load* pada *library Password_hash*. Sintaks pada baris selanjutnya akan melakukan
40 *hashing* pada *input* menggunakan algoritma yang ditentukan dan menyimpannya pada sebuah
41 variabel.

42 ***MY_Form_validation***

43 *MY_Form_validation* merupakan *library* yang dibentuk secara manual untuk menambahkan fungsi
44 validasi yang sudah tersedia pada *CodeIgniter 3*. *Library* ini memiliki dua buah fungsi yakni:

- **required**

46 Fungsi ini berguna untuk melakukan pengecekan terhadap sebuah *input* apakah berisikan
47 sebuah *array* kosong ataupun *string* kosong.

- **lowercase**

49 Fungsi ini berguna untuk melakukan pengecekan apakah *input* berisikan kata-kata dengan
50 huruf kecil atau tidak.

1 ***MY_Profiler***

2 *MY_Profiler* merupakan *library* yang dibentuk secara manual dan merupakan perpanjangan dari
 3 *CI_Profiler*. *Library* ini berfungsi untuk mengembalikan data yang telah dijalankan oleh *profiler*
 4 dan menyimpannya pada halaman pengguna. *Library* ini tidak akan digunakan lagi karena tidak
 5 terdapat pada *CodeIgniter 4* dan tidak dipakai pada *SharIF Judge*.

6 ***Parsedown***

7 *Parsedown* merupakan sebuah *library* yang dibentuk oleh Emanuil Rusev. *Library* ini berfungsi
 8 untuk mengubah teks dengan sintaks *markdown* menjadi teks dalam bentuk file lain seperti HTML.
 9 Kode 3.4 merupakan contoh penggunaan *library parsedown*.

Kode 3.4: Contoh kode penggunaan *Library Parsedown*

```
10
11  $this->load->library('parsedown');
12  $html = $this->parsedown->parse(file_get_contents("$assignment_dir/p$i/desc.md"));
```

14 Kode 3.4 akan menginisiasi *library parsedown* pada sintaks baris pertama. Selanjutnya isi dari
 15 file *desc.md* akan diambil dan dilakukan *parsedown* menjadi file HTML. Berikut merupakan contoh
 16 dari teks sebelum dan sesudah di *parsedown*:

17 Ini adalah list :

- 18 1. Satu
- 19 2. Dua
- 20 3. Tiga

Kode 3.5: Contoh kode sesudah dilakukan *parsedown*

```
21
22  <p>Ini adalah list</p>
23  <ol>
24  <li>Satu</li>
25  <li>Dua</li>
26  <li>Tiga</li>
```

28 Kode 3.5 merupakan contoh kode HTML sesudah dilakukan *parsedown*. Angka 1, 2, dan 3 akan
 29 diubah menjadi *list* yang dapat ditampilkan pada HTML.

30 ***Phpxcel***

31 *Phpxcel* merupakan sebuah *library* yang dibentuk oleh *PHPExcel*. *Library* ini memiliki fungsi untuk
 32 mengubah data yang ada pada PHP menjadi file excel.

33 ***Shj_pagination***

34 *Shj_pagination* merupakan *library* yang dibentuk secara manual dengan fungsi untuk membatasi
 35 maksimal data pada setiap halaman sesuai dengan konfigurasinya.

36 ***Upload***

37 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk menerima
 38 masukan dan mengunggah file yang dimasukan oleh pengguna. Kode 3.6 merupakan penggunaan
 39 *library upload* pada *SharIF Jugde*.

Kode 3.6: Contoh penggunaan *library upload*

```

1  $this->load->library('upload');
2  $config = array(
3      'upload_path' => $assignments_root,
4      'allowed_types' => 'zip',
5  );
6  $this->upload->initialize($config);
7  $zip_uploaded = $this->upload->do_upload('tests_desc');
8  $u_data = $this->upload->data();

```

11 Kode 3.6 merupakan contoh penggunaan *library upload* untuk menerima *input* berupa post.
 12 Sintaks baris pertama akan melakukan inisiasi terhadap *library upload*. Sedangkan sintaks selanjutnya akan menentukan dan menginisiasikan konfigurasi terhadap *path* dan tipe apa saja yang dapat dikirimkan oleh pengguna. Setelah itu akan dilakukan *upload* menggunakan sintaks *do_upload*.
 13 Terakhir akan disimpan seluruh data file yang telah dilakukan *upload* menuju variabel.

16 *Input*

17 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk menerima *input* dari pengguna. Kode 3.7 merupakan penggunaan *library input* pada *SharIF Judge*.

Kode 3.7: Contoh penggunaan *library input*

```

19 1   $this->load->library('upload');
20 2   $names = $this->input->post('name');

```

23 Kode 3.7 merupakan contoh penggunaan *library upload* untuk menerima *input* berupa post. Sintaks baris pertama akan melakukan inisiasi terhadap *library upload*. Sedangkan sintaks selanjutnya akan menerima data bernama *name* yang dikirim melalui *post*.

26 *Email*

27 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk mengirimkan *email* kepada orang yang dituju sesuai dengan konfigurasinya.

29 *Form validation*

30 *Library* ini merupakan fungsi yang tersedia pada *CodeIgniter 3*. *Library* ini berguna untuk melakukan validasi terhadap data yang dimasukan oleh pengguna sesuai dengan aturan yang telah ditetapkan.

32 *Kelas URI*

33 *Library* ini berfungsi untuk mengambil informasi yang terdapat pada *segment* dalam sebuah *URI*.
 34 Kode 3.8 merupakan penggunaan *library* ini untuk mengambil *segment* dan mengubahnya menjadi sebuah *array*.

Kode 3.8: Contoh penggunaan *library URI*

```

36 1   $input = $this->uri->uri_to_assoc();

```

39 Kode 3.8 merupakan fungsi untuk mengambil informasi yang tersedia pada *segment*. Hasil data dari pengambilan tersebut berupa *associative array*.

1 *Session*

2 *Library* ini berfungsi untuk menyimpan informasi setiap pengguna memakai perangkat lunak.
3 Data-data ini selanjutnya disimpan menuju *database* dengan nama tabel `shj_session`. Data-data
4 yang telah disimpan selanjutnya dapat diambil kembali untuk melakukan pengecekan ataupun
5 disimpan menuju tabel lainnya. Berikut merupakan penggunaan *library session* untuk mengambil
6 data pengguna:

Kode 3.9: Contoh penggunaan *library session*

7 1 `$this->session->userdata('logged_in')`

10 Kode 3.9 merupakan penggunaan *library* ini untuk mengambil data dengan nama `logged_in`.
11 *SharIF Judge* menggunakan tabel untuk menyimpan *session* dengan isi kolom berupa `session_id`,
12 `ip_address`, `user_agent`, `last_activity`, dan `user_data`.

13 **3.2 Analisis Sistem Usulan**

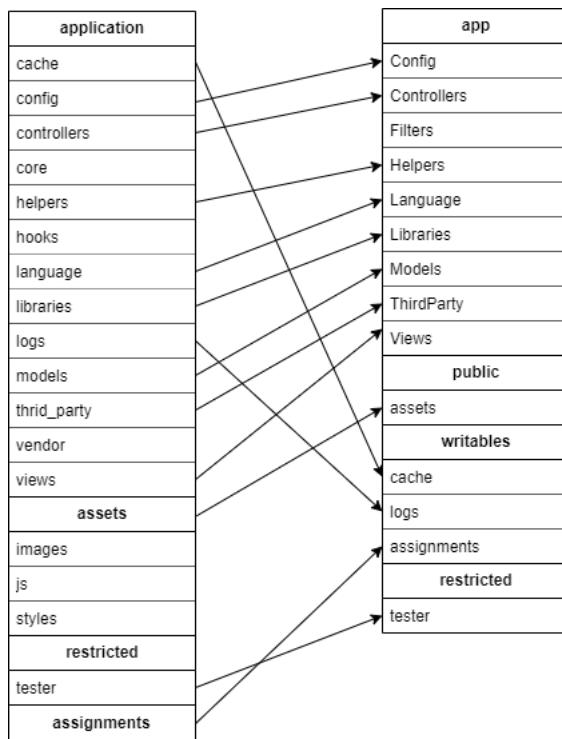
14 Konversi *CodeIgniter 3* menuju *CodeIgniter 4* diperlukan penulisan ulang karena terdapat perubahan
15 struktur aplikasi dan beberapa fungsi yang memiliki pemanggilan berbeda dan harus dilakukan
16 pembaharuan.

17 **3.2.1 Persiapan *CodeIgniter 4***

18 Konversi dimulai dengan mempersiapkan aplikasi *CodeIgniter 4* dengan mengunduh ataupun
19 memasangnya melalui *Composer*. Pengguna juga perlu memasang komponen pendukung seperti
20 `phppoffice`, `radius`, dan `adldap2`.

21 **3.2.2 Struktur Aplikasi**

22 Struktur aplikasi pada *CodeIgniter 3* dan *CodeIgniter 4* memiliki perubahan sehingga perlu dilakukan
23 pemindahan *file-file* menuju *CodeIgniter 4*. Gambar 3.17 merupakan pemindahan struktur aplikasi
24 *SharIF Judge* pada *CodeIgniter 3* menuju *CodeIgniter 4*.



Gambar 3.17: Pemindahan struktur aplikasi menuju *CodeIgniter 4*

1 Berikut merupakan rincian direktori yang akan dipindahkan menuju *CodeIgniter 4*.

2 Application

3 Direktori-direktori **application** pada *CodeIgniter 3* akan dipindahkan dengan penyesuaian menuju
4 direktori **app** terkecuali direktori **vendor**, **cache** dan **core**. Berikut merupakan direktori yang
5 dipindahkan dari direktori *application* menuju direktori *app*.

- 6 • **application/config** akan dipindahkan menuju **app/Config**.
- 7 • **application/controllers** akan dipindahkan menuju **app/Controllers**.
- 8 • **application/helpers** akan dipindahkan menuju **app/Helpers**.
- 9 • **application/language** akan dipindahkan menuju **app/Language**.
- 10 • **application/libraries** akan dipindahkan menuju **app/Libraries**.
- 11 • **application/models** akan dipindahkan menuju **app/Models**.
- 12 • **application/views** akan dipindahkan menuju **app/views**.

13 Config

14 Data dari file pada direktori **config** *CodeIgniter 3* akan dipindahkan dengan penyesuaian menuju
15 direktori **Config**. Berikut merupakan file-file yang dipindahkan dengan penyesuaian.

- 16 • **autoload.php** akan dipindahkan menuju **BaseController** seperti **helpers** dan terdapat peng-
hapusan **autoload library** dan **model** karena perbedaan cara kerjanya.
- 17 • **config.php** akan dipecah dan dipindahkan menjadi tiga buah file yakni **Config/app.php**,
Config/Session.php, dan **Config/Security.php**.
- 18 • **constants.php** akan dipindahkan menuju **Config/Constants.php**.

- 1 • `database.php` akan dipindahkan menuju `Config/Database.php`.
- 2 • `routes.php` akan dipindahkan menuju `Config/Routes.php`.
- 3 • `secrets.php` akan dibentuk baru pada direktori `Config`.

4 ***Public***

5 *CodeIgniter 3* tidak menyediakan direktori akar berupa `public` sehingga terdapat perubahan struktur
6 dimana direktori yang sebelumnya ada pada `application` akan dipindah menuju direktori `public`.
7 Berikut merupakan direktori yang dipindahkan menuju direktori `public`.

- 8 • `assets` akan dipindahkan menuju `public/assets`.

9 Selain struktur aplikasi diatas, *SharIF Judge* memiliki dua buah direktori terpisah diluar direktori
10 utama bernama `assignments` dan `tester`. Direktori `assignments` ini berfungsi untuk menyimpan
11 seluruh `file` yang telah dikumpulkan sedangkan direktori `tester` digunakan untuk melakukan perco-
12 baan untuk keamanan `sandbox`. Kedua direktori ini harus dapat ditulis oleh `PHP` sehingga akan
13 dipindahkan menuju direktori `writables`.

14 ***Writables***

15 Direktori ini merupakan direktori berisikan seluruh direktori yang dapat ditulis oleh `PHP`. Berikut
16 merupakan direktori yang dipindahkan menuju `writables`.

- 17 • `application/cache` akan dipindahkan menuju `writables/cache`.
- 18 • `application/log` akan dipindahkan menuju `writables/logs`.
- 19 • `assignments` akan dipindahkan menuju `writables/assignments`.

20 ***3.2.3 Routing***

21 *Routing* pada aplikasi *SharIF Judge* menggunakan `auto routing` yang telah disediakan oleh *CodeIg-
22 niter 3*. *Auto routing* akan membentuk `url` sesuai dengan `controller` dan `method` yang telah dibentuk
23 tanpa harus didefinisikan secara manual. Penggunaan `auto routing` seperti pada *CodeIgniter 3*
24 memiliki kekurangan pada bagian keamanan dimana `filter` pada `controller` dan proteksi `CSRF` akan
25 dilewati. Sehingga, konversi pada aplikasi *SharIF Judge* akan menggunakan `URI Routing` yang
26 didefinisikan secara manual untuk alasan keamanan dan `url` yang fleksibel. Berikut merupakan
27 contoh `route` yang didefinisikan secara manual.

28 \$routes->post("login/register", 'Login::register');

29 *Route* akan didefinisikan secara eksplisit sesuai dengan fungsi dan metodenya karena alasan keamanan
30 dan juga konfigurasi `filters` yang berbeda pada setiap `routes`.

31 ***3.2.4 Model, View, and Controller***

32 *CodeIgniter 4* memiliki perubahan baik dari kegunaan dan cara pemanggilan `Model, View, and
33 Controller`. Berikut merupakan perubahan yang terjadi:

1 *Model*

2 *Model* pada *CodeIgniter 4* memiliki perubahan dimana *model* dapat digunakan untuk mengambil
 3 data pada satu buah tabel spesifik. Konversi *model* dari *CodeIgniter 3* menuju *CodeIgniter 4* dapat
 4 dilakukan menggunakan dua buah cara yakni:

- 5 1. Menggunakan *Model* dari *CodeIgniter 3*

6 *Model* akan dipindahkan menuju direktori `app/Models` dan dilakukan beberapa perubahan.
 7 Kode 3.10 merupakan contoh fungsi yang terdapat pada *CodeIgniter 3*.

Kode 3.10: Contoh fungsi untuk mengambil data seluruh user pada *CodeIgniter 3*

```
8
9 1 public function get_all_users()
10 2 {
11 3     return $this->db->order_by('role', 'asc')->order_by('id')->get('users')->result_array();
12 4 }
```

14 Kode 3.10 mengambil data dari tabel `users` dengan hasil berupa *array* dan diurutkan sesuai
 15 dengan `role` dan `id`nya. Kode akan dilakukan konversi dengan pengubahan sintaks. Berikut
 16 merupakan hasil konversi fungsi untuk mengambil data seluruh *user*.

Kode 3.11: Hasil konversi fungsi untuk mengambil data seluruh user

```
17
18 1 public function get_all_users()
19 2 {
20 3     return $this->db->table('users')->orderBy('role', 'asc')->orderBy('id')->get()->getResultArray();
21 4 }
```

23 Kode 3.11 merupakan hasil konversi *model SharIF Judge* dari *CodeIgniter 3* menjadi *CodeIgniter 4*. Pada *CodeIgniter 4* diperlukan untuk memberikan nama tabel yang akan diakses dan
 24 juga terdapat perubahan sintaks menjadi *camelCase*.

- 26 2. Menggunakan *Model* pada *CodeIgniter 4*

27 *CodeIgnier 4* menyediakan fungsi *model* yang dapat dibentuk melalui *command line* untuk
 28 sebuah tabel spesifik. Pengguna dapat membentuk *model* melalui *command line* menggunakan
 29 sintaks sebagai berikut.

30 `make:model <name>`

31 *Model* pada *CodeIgniter 4* menyediakan fungsi untuk mengambil, memasukan, dan memper-
 32 baharui data dari sebuah tabel spesifik tanpa harus membentuk secara manual fungsi-fungsi
 33 tersebut. Pengguna dapat melakukan inisiasi dan memakai fungsi untuk mengambil data
 34 menggunakan sintaks berikut.

35 `$userModel = new \App\Models\UserModel();`

36 `$user = $userModel->findAll();`

37 Sintaks diatas akan mengambil seluruh data dari `$userModel` sesuai dengan konfigurasi yang
 38 telah dilakukan pengguna. Pengguna juga dapat melakukan *create*, *update*, dan *delete* melalui
 39 sintaks berikut.

40 `$this->Notifications_model->delete('notifications', array('id' => $id));`

41 Sintaks diatas akan menghapus data dari tabel `notifications` sesuai dengan `id` yang diberikan.
 42 Fungsi ini tidak perlu membentuk sintaks secara manual pada file *models* melainkan hanya perlu

1 memanggil *model* yang ingin digunakan. Konversi aplikasi *SharIF Judge* akan menggunakan cara
 2 pertama dengan penghapusan dan pembaharuan beberapa sintaks. Selain itu, terdapat perubahan
 3 nama *model* dari yang sebelumnya menggunakan *snake case* menjadi *PascalCase* agar sesuai dengan
 4 standar PHP.

5 *View*

6 *View* pada aplikasi *SharIF Judge* menggunakan *template engine* bernama *Twig*. *Twig* merupakan
 7 sebuah *template engine* untuk bahasa pemrograman *PHP* yang berguna untuk mempermudah
 8 dalam membentuk tampilan sebuah aplikasi. *Twig* tidak terintegrasi pada *CodeIgniter 4* sehingga
 9 akan terdapat beberapa pemasangan dan perubahan pada sintaks yang telah dipasang. Selain itu,
 10 terdapat beberapa perubahan fungsi pada *CodeIgniter 4* sehingga perlu dilakukan penyesuaian
 11 seperti pengubahan *file extension* dari *.twig* menjadi *.php*. Konversi *SharIF Judge* menuju
 12 *CodeIgniter 4* akan mengubah *view* yang sebelumnya menggunakan *twig* menjadi menggunakan
 13 *PHP* sesuai dengan dokumentasi *CodeIgniter 4*. Seluruh *delimiters* akan diubah menggunakan fitur
 14 yang terdapat pada *CodeIgniter 4*. Kode 3.12 merupakan contoh konversi yang dilakukan.

Kode 3.12: Contoh *view* menggunakan *twig*

```
15 15 {% if registration_enabled %}  
16 16     <a href="{{ site_url('register') }}>Register</a> |  
17 17     {% endif %}
```

20 menjadi kode berikut:

Kode 3.13: Contoh *view* menggunakan *php*

```
21 21 <?php if($registration_enabled): ?>  
22 22     <a href=<?= site_url('register') ?>>Register</a> |  
23 23     <?php endif ?>
```

26 Seluruh sintaks *twig* akan diubah menjadi sintaks *PHP* dari *CodeIgniter 4* seperti yang terdapat
 27 pada kode 3.13.

28 *Controller*

29 *Controller* pada *CodeIgniter 4* memiliki fungsi sama dengan pada *CodeIgniter 3* sehingga hanya
 30 akan dipindahkan dan dilakukan penghapusan dan perubahan pada sintaks yang ada. Sintaks
 31 yang dihapus berupa *defined* dan akan digantikan dengan *namespace*. Beberapa sintaks untuk
 32 memanggil *models* dan pengembalian *view* juga akan disesuaikan. Namun, terdapat perubahan
 33 pada *constructor* dimana pada *CodeIgniter 4* terdapat *initController*. *Constructor* pada *PHP*
 34 tidak diperbolehkan untuk mengembalikan apapun sehingga terdapat beberapa pemindahan fungsi
 35 seperti *redirect()* menuju *filters*. Selain itu, konversi akan tetap menggunakan *__construct*
 36 dengan pemindahan beberapa sintaks menuju *initController* seperti pemanggilan *helpers* dan
 37 variabel yang dapat diakses pada seluruh *controller*. Berikut merupakan contoh penggunaan
 38 *initController* untuk melakukan inisiasi fungsi menuju variabel.

Kode 3.14: Penggunaan *initController* untuk inisiasi fungsi menuju variabel

```
39 39 protected $config;  
40 40  
41 41 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)  
42 42 {  
43 43     // Do Not Edit This Line
```

```

16 parent::initController($request, $response, $logger);
17
18 // Preload any models, libraries, etc, here.
19
20 // E.g.: $this->session = \Config\Services::session();
21 $this->config = Config('Secrets');
22 }

```

9 Kode 3.14 akan memanggil variabel `$config` berisikan data dari file `config Secrets.php` yang
10 dapat digunakan seluruh *controller* tanpa perlu melakukan inisiasi lagi.

11 Fungsi lainnya akan dipindahkan sesuai dengan yang ada pada *CodeIgniter 3* dengan pembaharuan. Selain pembaharuan, akan terdapat pemindahan variabel *global* yang sebelumnya telah
13 diinisiasi menuju *controller*.

14 IncomingRequest

15 *Input* akan digantikan menggunakan fungsi *request* dengan perubahan beberapa fungsi. Kode 3.15
16 merupakan perubahan yang terdapat dalam menerima *input* dari pengguna.

Kode 3.15: Contoh perubahan *library request*

```

17 $this->request = \Config\Services::request();
18 $names => $this->request->getPost('name'),

```

21 Kode 3.15 merupakan perubahan dalam menerima data yang dikirim melalui *post*. Sintaks pertama
22 akan melakukan inisiasi terhadap *library* tersebut namun inisiasi tersebut hanya diperlukan pada
23 *model* karena fungsi ini sudah terdapat pada *controller*. Sintaks baris kedua akan mengambil data
24 bernama *name* yang telah dikirimkan oleh pengguna melalui *post*. Konversi aplikasi *SharIF Judge*
25 akan menggunakan fungsi ini dengan beberapa perubahan sintaks sesuai dengan dokumentasinya.

26 3.2.5 Libraries

27 *Libraries* pada *CodeIgniter 3* memiliki perubahan dan penghapusan pada *CodeIgnier 4* sehingga
28 perlu dilakukan pembaharuan. Berikut merupakan *libraries* yang dipakai pada *SharIF Judge*:

29 Emails

30 *Emails* pada *CodeIgnier 4* terdapat perubahan sintaks dan cara pemanggilan sehingga akan dipindahkan
31 sesuai dengan sintaks yang baru. Sintaks berubah dari yang sebelumnya menggunakan
32 *snake case* menjadi menggunakan *camelcase*. Kode 4.24 merupakan contoh penggunaan *library*
33 *email*.

Kode 3.16: Contoh perubahan *library emails*

```

34 $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
35 $this->email->setTo($user[1]);
36 $this->email->setSubject('SharIF Judge Username and Password');
37 $text = $this->settings_model->get_setting('add_user_mail');
38 $text = str_replace('{SITE_URL}', base_url(), $text);
39 $text = str_replace('{ROLE}', $user[4], $text);
40 $text = str_replace('{USERNAME}', $user[0], $text);
41 $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
42 $text = str_replace('{LOGIN_URL}', base_url(), $text);
43 $this->email->setMessage($text);
44 $this->email->send()

```

47 Kode 4.24 memiliki sintaks dengan nama sama namun terdapat perubahan menjadi *camelcase*.

1 ***Working with Uploaded Files***

2 *Upload* akan digantikan dengan fungsi *Working with uploaded files* dengan beberapa penggantian
 3 dan penghapusan fungsi. *Working with uploaded files* terdapat perubahan pada beberapa sintaks
 4 dan validasi terhadap *file* yang telah diunggah. Kode 3.17 merupakan perubahan yang terdapat
 5 dalam melakukan *upload* sebuah file.

Kode 3.17: Contoh perubahan *library upload*

```
6 1 $zip_uploaded = $this->request->getFile('tests_desc');
7 2 $pdf_uploaded->move("$assignment_dir");
```

10 Kode 3.17 merupakan perubahan dalam melakukan *upload* sebuah file. Sintaks *getFile* akan
 11 mengambil file bernama *tests_desc* yang dikirimkan oleh pengguna. Sintaks selanjutnya akan
 12 memindahkan file tersebut menuju direktori yang telah ditentukan. Pada *CodeIgniter 4* sudah
 13 tidak terdapat konfigurasi aturan dan pengambilan data terhadap file yang diterima. Sehingga
 14 fungsi tersebut akan digantikan oleh *validation* dan beberapa fitur yang terdapat pada *library ini*.
 15 Berikut merupakan contoh sintaks untuk mengambil *extension* dan data dari file yang diunggah.

Kode 3.18: Contoh sintaks untuk mengambil *extension* dan data file

```
16 1 if($pdf_uploaded->getExtension() != "pdf"){
17 2   $zip_uploaded->getName()
18 3 }
```

21 Kode 3.18 merupakan contoh sintaks untuk melakukan pengecekan *extension* dan pengambilan
 22 data berupa nama dari file.

23 ***Working with URIs***

24 Kelas *URI* akan digantikan menggunakan fungsi *working with URIs* dengan perubahan fungsi.
 25 Berikut merupakan contoh pengambilan sebuah *segment* yang terdapat pada *URI*:

```
26 $temp_inp = $this->uri->getSegments();
```

27 Sintaks diatas akan mengambil semua *segment* yang tersedia pada *URI*. *Segment* yang sudah diambil
 28 akan disimpan menjadi sebuah *array*.

29 ***Session***

30 Pemakaian *library* ini terdapat perubahan dalam sintaks dan inisiasi *library*. *Library* ini perlu
 31 dilakukan inisiasi pada *CodeIgniter 4*. Berikut merupakan contoh inisiasi *library* ini:

```
32 $this->session = session();
```

33 Sintaks diatas akan melakukan inisiasi terhadap *library* yang nantinya dapat dipakai pada seluruh
 34 file. Selanjutnya terdapat perubahan cara pengambilan data berikut merupakan contoh pengambilan
 35 data pada *CodeIgniter 4*:

```
36 $this->session->get('logged_in')
```

37 Sintaks diatas akan mengambil data dengan nama *logged_in*. Penggunaan berubah dari yang
 38 sebelumnya menggunakan sintaks *userdata*. Selanjutnya data tersebut akan disimpan menuju
 39 *database* dengan perubahan dan penghapusan kolom. Kolom tabel akan berubah menjadi empat
 40 buah yakni *id*, *ip_address*, *timestamp*, dan data.

1 Validation

2 *Form_validation* akan digantikan menggunakan fungsi *validation* dengan perubahan dan pengapusan
 3 beberapa fungsi. Berikut merupakan contoh penetapan aturan untuk melakukan validasi saat
 4 mengumpulkan sebuah data pada *form*:

```
5         $validate->setRule('username', 'username',
6             'required|min_length[3]|max_length[20]');
```

7 Sintaks diatas akan melakukan validasi terhadap *input* yang akan masukan oleh pengguna. Sintaks
 8 penetapan aturan berubah menggunakan *camelCase* dari yang sebelumnya berupa *set_rules*
 9 menjadi *setRule*. Aturan untuk validasi tetap menggunakan fungsi yang tersedia pada *CodeIgniter*
 10 tanpa perubahan namun, terdapat pemindahan aturan yang dibentuk manual dan penambahan
 11 beberapa sintaks. Kode 3.19 merupakan pemindahan aturan dengan penambahan beberapa sintaks:

Kode 3.19: Contoh perubahan penambahan sintaks pada aturan yang dibentuk manual

```
12
13 1     public function email_check($email, $edit_username):bool
14 2     {
15 3         $edit_username = explode(',', $edit_username);
16 4         $user_model = new UserModel();
17 5         if ($user_model->have_email($email, $edit_username[1]))
18 6             return FALSE;
19 7         return TRUE;
20 8     }
```

22 Kode 3.19 merupakan pemindahan aturan menuju *app/Config/Validation.php* aturan akan
 23 ditambahkan sintaks *explode* untuk memecah *array* dan juga terdapat inisiasi *model*. Aturan ini da-
 24 pat digunakan seperti aturan lainnya. Namun, *CodeIgniter 4* tidak menyediakan fungsi *form_error*
 25 sehingga akan diubah dengan menggunakan fungsi baru bernama *validation_errors()*. Fungsi
 26 tersebut dapat digunakan untuk mengembalikan *error* apabila terdapat data yang tidak sesuai
 27 dengan aturan. *Error* tersebut dapat ditampilkan pada halaman *view* menggunakan sintaks berikut.

```
28 <?= $validation->getError('username'); ?>
```

29 Sintaks diatas akan mengembalikan *error* terhadap *form* dengan nama *username* apabila tidak
 30 sesuai dengan aturan yang sudah ditentukan. Variabel *validation* akan dikirimkan dari *controller*
 31 berisikan *library* dari *validation* tersebut.

32 Zip Archive

33 *Zip Encoding* dan *Unzip* akan digantikan dengan fungsi PHP *zip archive* karena sudah tidak tersedia
 34 pada *CodeIgniter 4*. Fungsi *zip archive* terdapat beberapa perbedaan sehingga akan disesuaikan
 35 dengan fungsi-fungsi yang ada. Berikut merupakan penggunaan *zip archive* untuk melakukan *zip*.

Kode 3.20: Contoh perubahan penggunaan *library Zip Archive* untuk melakukan *zip*

```
36
37 1     $this->zip = new \ZipArchive();
38 2     $this->zip->open($zipname, ZipArchive::CREATE);
39 3     $this->zip->addFile($file_path, "{$item['username']}/p{$item['problem']}." . filetype_to_extension($item['file_type']));
```

41 Kode 3.20 merupakan perubahan dalam melakukan *zip* sebuah file. Baris pertama akan
 42 melakukan inisiasi terhadap *ZipArchive* sedangkan baris kedua akan membuka file *zip*. Terakhir
 43 sintaks *addFile* akan memasukan file bernama *file_path* menuju zip yang terdapat pada *array item*.

- 1 Selain untuk membentuk file *zip*, *zip archive* juga dapat melakukan *unzip* terhadap file. Kode 3.21
 2 merupakan penggunaan *ZipArchive* dalam melakukan *unzip*.

Kode 3.21: Contoh perubahan penggunaan *library ZipArchive* untuk melakukan *unzip*

```

3 1 $this->unzip = new ZipArchive();
4 2     // Create a temp directory
5 3     $tmp_dir = "$assignments_root/$tmp_dir_name";
6 4     // Extract new test cases and descriptions in temp directory
7 5     $this->unzip->open("$assignments_root/".$zip_uploaded->getName());
8 6     $extract_result = $this->unzip->extractTo($tmp_dir);

```

11 Kode 3.21 merupakan perubahan dalam melakukan *unzip* dalam *ZipArchive*. Baris pertama
 12 akan melakukan inisiasi terhadap *ZipArchive* sedangkan baris kedua membentuk sebuah direktori
 13 sementara bernama *assignments_root/shj_tmp_directory*. Baris ketiga akan membuka sebuah *zip*
 14 pada *assignments_root* dengan nama yang dibentuk. Baris terakhir akan melakukan *extract* file
 15 pada direktori yang sudah dibentuk.

16 ***Twig***

17 *Library* ini tidak akan digunakan untuk membentuk *view* pada *CodeIgniter 4* namun, akan ada
 18 penggunaan sebuah fungsi *Twig* yang akan dibentuk pada direktori *app/Libraries*. Fungsi tersebut
 19 bernama **extra_time_formatter** yang memiliki fungsi untuk mengubah input yang diberikan
 20 menjadi format jam dikali enam puluh menit.

21 ***Unzip***

22 *Library* ini tidak akan digunakan dan digantikan oleh *ZipArchive* yang disediakan oleh PHP. Terdapat
 23 perubahan sintaks pada *ZipArchive* sehingga akan disesuaikan. Perubahan dapat dilihat pada
 24 subbab 3.2.5 kode 3.21.

25 ***Password_hash***

26 *Library* ini tidak akan digunakan dan akan digantikan oleh *password hash* yang disediakan oleh
 27 PHP .*Library Password_hash* merekomendasikan pegguna untuk menggunakan fungsi *native* yang
 28 disediakan oleh PHP apabila aplikasi mendukung PHP versi 5.5 ke atas. Sehingga, akan dilakukan
 29 konversi menggunakan fungsi yang disediakan oleh PHP bernama *password_hash()*. Seluruh
 30 penggunaan *library* ini akan diubah menggunakan fungsi yang disediakan oleh PHP dengan metode
 31 *hashing* sama yaitu *CRYPT_BLOWFISH*. Perubahan fungsi *hashing* ini bersifat *backward compatible*
 32 sehingga dapat menggunakan *database* aplikasi terdahulu tanpa perlu membentuk data baru. Berikut
 33 merupakan contoh pengubahan kode dari *phpass* menjadi *password_hash*.

34 'password' => \$this->password_hash->HashPassword(\$password)

35 menjadi

36 'password' => password_hash(\$password,PASSWORD_BCRYPT)

37 Sintaks *password_hash()* diatas menerima dua buah parameter yakni data yang ingin di enkripsi
 38 dan tipe enkripsi. Enkripsi akan menggunakan sintaks *PASSWORD_BCRYPT* yang menggunakan tipe
 39 *hash* berupa *CRYPT_BLOWFISH*.

1 MY_Form_validation

2 *Library MY_Form_validation* akan dipindahkan menuju direktori `app/Libraries`. *Library* ini
 3 akan digunakan kembali dengan perubahan `extends` menjadi menuju `Validation`, penghapusan
 4 sintaks `defined`, dan akan ada penambahan `namespace` pada baris awal file. Kode 3.22 merupakan
 5 contoh penambahan `namespace` dan penggantian `extends` pada *library* ini.

Kode 3.22: Contoh perubahan *library MY_Form_validation* pada *CodeIgniter 4*

```
6 namespace App\Libraries;
7 1
8 2
9 3 use CodeIgniter\Validation\Validation;
10 4
11 5 class MY_Form_validation extends Validation
```

13 Kode 3.22 menghapus sintaks `defined` dan menggantikannya dengan penambahan `namespace`. Selain
 14 itu, kelas *library* akan `extends Validation`.

15 Parsedown

16 *Library Parsedown* akan dipindahkan menuju direktori `app/Libraries`. *Library* ini akan digunakan
 17 kembali dengan penambahan `namespace` pada baris awal file dan penghapusan sintaks `defined`.
 18 Kode 4.36 merupakan contoh penambahan `namespace` dan juga penambahan sintaks `defined`.

Kode 3.23: Contoh perubahan *library Parsedown* pada *CodeIgniter 4*

```
19 namespace App\Libraries;
20 1
21 2
22 3 class Parsedown
```

24 Kode 4.36 menghapus sintaks `defined` dan menggantikannya dengan penambahan `namespace`.

25 Phpexcel

26 *Library* ini akan digunakan kembali namun tidak akan dipindahkan menuju `app/Libraries`. *Library*
 27 akan dilakukan instalasi melalui `composer` dengan sintaks berikut:

```
28 composer require phoffice/phpexcel
```

29 Sintaks diatas akan dijalankan pada akar dari aplikasi dan tidak terdapat perubahan terhadap
 30 penggunaan sintaks ini.

31 Shj_pagination

32 *Library* ini akan digunakan kembali dan dipindahkan menuju direktori `app/Libraries`. Selain itu,
 33 terdapat penambahan `namespace` pada baris awal file dan penghapusan sintaks `defined`.

34 3.2.6 Configuration

35 *Configuration* terdapat perubahan nama dari yang sebelumnya `application/config/config.php`
 36 menjadi `app/Config/App.php` dan penambahan `file` dengan nama `app/Config/Secrets.php`. Ber-
 37 hubung dengan perubahan nama tersebut, terdapat beberapa perpindahan sintaks menuju direktori
 38 baru tersebut. Berikut merupakan sintaks yang dipindahkan menuju `app/Config/Security.php`:
 39

- `$config['csrf_protection'] = TRUE;`

- 1 • \$config['csrf_token_name'] = 'shj_csrf_token';
- 2 • \$config['csrf_cookie_name'] = 'shjcsrftoken';
- 3 • \$config['csrf_expire'] = 7200;
- 4 • \$config['csrf_regeneration'] = FALSE;

5 Configurations yang telah dipindahkan akan diubah dari yang sebelumnya menggunakan *array*
 6 menjadi menggunakan *variable*. Seluruh configurations pada *CodeIgniter 3* akan dipindahkan
 7 menuju *CodeIgniter 4* sesuai dengan direktori dan fungsinya. Sedangkan app/Config/App.php
 8 dan dipindahkan menuju file .env karena alasan kemudahan untuk penggantian *environment* dalam
 9 melakukan *deploy*.

10 3.2.7 Database

11 Database pada *CodeIgniter 4* memiliki fungsi yang sama pada *CodeIgniter 3* sehingga akan dilakukan
 12 pemindahan konfigurasi sesuai dengan yang ada pada *CodeIgniter 3*. Namun, terjadi beberapa
 13 perubahan pada sintaks untuk melakukan koneksi ke *database* dan beberapa sintaks untuk melakukan
 14 *query*. Sintaks koneksi *database* akan berubah dari \$this->load->database(); menjadi db =
 15 db_connect(). Selain itu, pemanggilan fungsi *query builder* berubah menggunakan *camelcase* dari
 16 yang sebelumnya menggunakan *snakecase*. Seperti dari yang sebelumnya menggunakan sintaks
 17 get_where menjadi getWhere.

18 Database pada aplikasi *SharIF Judge* menggunakan *autoload* yang dapat memuat beberapa
 19 fungsi secara otomatis. Namun, pada *CodeIgniter 4* tidak akan menggunakan *autoload* karena
 20 terdapat perbedaan fungsi *autoload* pada *CodeIgniter 4* sehingga akan dimuat menggunakan \$db =
 21 db_connect() pada *controller* yang membutuhkan.

22 3.2.8 Helpers

23 Helpers tidak terdapat banyak perubahan namun, terdapat perubahan pemanggilan *helpers* dan
 24 beberapa penghapusan *helpers*. Berikut merupakan *helpers* yang dihapus dan diubah cara pemang-
 25 gilannya.

- 26 • Download Helper Helper ini sudah tidak tersedia pada *CodeIgniter 4* sehingga perlu dilakukan
 27 pengubahan dengan menggunakan fungsi *HTTP Response*. *HTTP Response* menyediakan
 28 fungsi bernama *Force File Download* yang berguna untuk mengunduh sebuah *file* menuju
 29 perangkat pengguna. Fungsi ini dapat dipanggil menggunakan sintaks berikut.

```
30                  return $this->response->download($name, $data);
```

31 Sintaks diatas mengembalikan *file* yang ingin diunduh pengguna dengan dua buah parame-
 32 ter. Parameter pertama berupa nama *file* yang ingin diunduh sedangkan parameter kedua
 33 merupakan data dalam *file* tersebut.

- 34 • redirect()

35 Fungsi ini memiliki perubahan pada *CodeIgniter 4* dimana redirect() tidak langsung meng-
 36 arahkan kepada *url* yang dibentuk. Pengguna harus mengembalikan fungsi redirect() menggu-
 37 nakan return dengan sintaks sebagai berikut.

```
38                  return redirect()->to('login/form')
```

- 1 Sintaks diatas akan mengembalikan pengguna menuju *url login/form* yang sudah dibentuk
- 2 pada *routes*.
- 3 Konversi *helpers* akan dipindahkan dari yang sebelumnya menggunakan *autoload* menuju
- 4 *initController* dengan menambahkan pada variabel *helpers*.

1

BAB 4

2

PERANCANGAN

3 Bab ini membahas perancangan untuk seluruh implementasi *SharIF Judge* pada *CodeIgniter 4*.

4 **4.1 Instalasi *CodeIgniter 4***

5 *CodeIgniter 4* diinstalasi menggunakan *composer*. *Composer* merupakan sebuah *dependency manager* untuk PHP yang memungkinkan pengguna untuk melakukan instalasi seluruh kebutuhan untuk menjalankan program berbasis PHP. Instalasi akan dilakukan menggunakan sintaks sebagai berikut:

6

```
composer create-project codeigniter4/appstarter SharIF-JudgeCI4
```

7 Sintaks diatas akan mengunduh dan melakukan instalasi *template* projek *CodeIgniter 4*. Template berisikan *skeleton* dari projek *CodeIgniter 4* yang berisikan data untuk melakukan *development* sebuah aplikasi.

8 **4.2 Perubahan Struktur Aplikasi**

9 Struktur aplikasi *SharIF Judge* akan dipindahkan seperti pemetaan pada bab 3 gambar 3.17.
10 Pemindahan akan dilakukan dengan penyesuaian dan perubahan beberapa sintaks dan fungsi yang
11 digunakan. Struktur aplikasi pada *CodeIgniter 4* berisikan sebagai berikut :

12 **4.2.1 app/Config**

13 File *config* pada *CodeIgniter 3* dipindahkan sesuai dengan pemetaan pada gambar 3.17. Direktori
14 berisikan data-data pada *application/Config*. Beberapa data pada direktori ini dipindahkan
15 menuju file yang terdapat pada *CodeIgniter 4*. Terdapat juga penambahan *file Secrets.php* yang
20 dibentuk secara manual. Berikut merupakan rincian isi direktori yang dipindahkan:

21 **app/Config/App.php**

22 File ini tidak akan digunakan sehingga akan dibiarkan berisi sintaks *default* dan seluruh data
23 akan dipindahkan menuju file *.env*. Kode 4.1 merupakan isi dari file *.env* yang dipindahkan dari
24 direktori *application/config.php*:

Kode 4.1: Kode *application/config/config.php* yang dipindahkan menuju *.env*

25

26

```
app.baseURL = 'http://sharif.localhost/'
```

1 Kode 4.1 akan menentukan *url* dasar dari aplikasi menjadi `http://sharif.localhost/`. *URL*
 2 dibentuk spesifik berupa `http://sharif.localhost/` karena dijalankan pada server lokal sehingga
 3 tidak bertabrakan dengan server lokal lainnya. *URL* juga diganti karena tidak dapat diimplementa-
 4 tasikan pada *CodeIgniter 4* sehingga *URL* akan ditentukan manual sesuai dengan *domain* yang
 5 dibentuk.

6 **app/Config/AutoLoad.php**

7 File ini tidak digunakan dan dibiarkan menggunakan konfigurasi *default* karena terdapat perubahan
 8 cara kerja kelas ini. Fitur ini pada *CodeIgniter 4* tidak dapat melakukan inisiasi otomatis terhadap
 9 *model*, *controller*, dan *library* saat aplikasi dijalankan. Namun kelas ini hanya akan mencari
 10 lokasi file sesuai dengan konfigurasinya saat kelas tersebut pertama kali diinisiasi. Kelas *helpers*
 11 yang diinisiasikan dipindahkan menuju *BaseController* sedangkan kelas *model* dan *libraries* akan
 12 dipanggil menggunakan *PSR-4* pada file-file yang membutuhkan kelas tersebut.

13 **app/Config/Cache.php**

14 File tidak berubah dan akan tetap menggunakan konfigurasi *default* karena tidak terdapat perubahan
 15 pada *SharIF Judge* versi *CodeIgniter 3*.

16 **app/Config/Constants.php**

17 File ini berisikan seluruh data yang dipindahkan dari `application/config/constants.php`. Kode
 18 4.2 merupakan data yang dipindahkan menuju `Constant.php`.

Kode 4.2: Pemindahan kode pada *Constant*

```
19 define('SHJ_VERSION', '1.4');

20 21 /*Code editor related constants*/
22 23 define('EDITOR_FILE_NAME', "editor");
24 25 define('EDITOR_FILE_EXT', "txt");
25 26 define('EDITOR_IN_NAME', "exec_in");
26 27 define('EDITOR_OUT_NAME', "exec_out");
27 28 define('EDITOR_SUBMIT_ID', 0);
```

29 Kode 4.2 merupakan pemindahan dari `application/config/constants.php` menuju `Constant.php`.
 30 Sintaks SHJ_VERSION akan digunakan pada tampilan sidebar sedangkan sintaks EDITOR digunakan
 31 pada fungsi submit. Sintaks yang telah dipindahkan akan bersifat *global* sehingga dapat diakses
 32 oleh seluruh file pada *CodeIgniter 4*.

33 **app/Config/Cookie.php**

34 File ini tidak terdapat perubahan dan tetap menggunakan konfigurasi *default* karena tidak terdapat
 35 perubahan pada *SharIF Judge* versi *CodeIgniter 3*.

36 **app/Config/Database.php**

37 File ini dibiarkan menggunakan konfigurasi *default* karena akan dipindahkan menuju file `.env`. Kode
 38 4.3 merupakan pemindahan menuju file `.env` yang dipindahkan dari direktori `application/config.php`.

Kode 4.3: Pemindahan app/config/database.php menuju .env

```

1 database.default.hostname = db
2 database.default.database = sharif
3 database.default.username = root
4 database.default.password =
5 database.default.DBDriver = MySQLi
6 database.default.DBPrefix = shj_
7

```

9 Kode 4.3 merupakan pemindahan kode dari `database.php` menuju `.env`. Kode akan berisikan
10 `hostname` yang digunakan, nama `database` yang digunakan, `username` dari `database`, `password`,
11 `DBDriver`, dan `DBPrefix` yang digunakan.

12 app/Config/Email.php

13 File ini tidak digunakan dan dibiarkan menggunakan konfigurasi `default` karena konfigurasi `email`
14 pada aplikasi *SharIF Judge* diletakan pada file `Secrets.php`.

15 File ini akan berisikan seluruh data yang dipindahkan dari `application/config/secrets.php`.
16 Kode 4.4 merupakan pemindahan konfigurasi `email`.

Kode 4.4: Pemindahan app/config/secrets.php menuju Email.php

```

17 public string $protocol = 'smtp';
18 public string $SMTPHost = 'ssl://smtp.mailgun.org';
19 public string $mailType = 'html';
20

```

22 Kode 4.4 merupakan pemindahan konfigurasi `email` yang terdapat pada `secrets.php`. Kode
23 akan berisikan `protocol` yang digunakan untuk mengirim `email`, `host` yang akan digunakan, dan
24 tipe `email` yang dikirimkan. Seluruh data akan disimpan menuju variabel pada *CodeIgniter 4* dan
25 terdapat perubahan nama variabel menjadi *camelCase*.

26 app/Config/Encryption.php

27 File ini akan berisikan seluruh data yang dipindahkan dari `application/config/config.php`.
28 Kode 4.5 merupakan pemindahan konfigurasi enkripsi yang akan digunakan.

Kode 4.5: Pemindahan app/config/config.php menuju Encryption.php

```

29 public string $key = 'bj42CAiqTwLHUhz3p0vfRM0EJFeQD9uG';
30

```

32 Kode 4.5 merupakan pemindahan `key` dari enkripsi yang akan digunakan. Data akan disimpan
33 menuju variabel pada *CodeIgniter 4*.

34 app/Config/Filters.php

35 File ini berisikan seluruh nama `filters` yang telah dibentuk. Kode 4.6 merupakan pembentukan
36 nama `filters` sesuai dengan kelas yang sudah dibentuk.

Kode 4.6: Penambahan nama `filters` untuk didefinisikan menuju routes

```

37 public array $aliases = [
38     'csrf'          => CSRF::class,
39     'toolbar'       => DebugToolbar::class,
40     'honeypot'      => Honeypot::class,
41     'invalidchars'  => InvalidChars::class,
42     'secureheaders' => SecureHeaders::class,
43     'check-installandlogin' => CheckInstallAndLogin::class,
44     'check-login'   => CheckLogin::class,
45     'check-loginandlevelAdmin' => CheckLoginAndLevelAdmin::class,
46     'check-loginandlevelHead' => CheckLoginAndLevelHead::class,
47

```

```

11     'check-loginandcli' => CheckLoginandCLI::class,
12     'check-loginandisajax' => CheckLoginandAjax::class,
13     'check-iscli' => CheckCLI::class,
14 ]

```

6 Kode 4.6 merupakan penambahan nama *filters* untuk dipanggil menuju *routes*. Penamaan ini
7 akan ditambahkan pada *array \$aliases* dengan nama lain *filters* dan nama kelasnya.

8 app/Config/Routes.php

9 File ini akan berisikan seluruh pemindahan dan penambahan manual dari *routes* pada *SharIF Judge*.
10 Kode 4.7 merupakan pembentukan *routes* aplikasi secara manual.

Kode 4.7: Penambahan *routes* yang digunakan pada aplikasi *SharIF Judge*

```

11 $routes->setDefaultNamespace('App\Controllers');
12 $routes->setDefaultController('Dashboard');
13 $routes->setDefaultMethod('index');
14 $routes->setTranslateURIDashes(false);
15 $routes->set404Override();
16 $routes->setAutoRoute(false);
17 // The Auto Routing (Legacy) is very dangerous. It is easy to create vulnerable apps
18 // where controller filters or CSRF protection are bypassed.
19 // If you don't want to define all routes, please use the Auto Routing (Improved).
20 // Set '$autoRoutesImproved' to true in 'app/Config/Feature.php' and set the following to true.
21 // $routes->setAutoRoute(false);
22
23 /*
24 * -----
25 * Route Definitions
26 * -----
27 */
28
29 // We get a performance increase by specifying the default
30 // route since we don't have to scan directories.
31 $routes->get('/', 'Dashboard::index', ['filter' => 'check-installandlogin:dual,noreturn']);
32 $routes->get('/dashboard', 'Dashboard::index', ['filter' => 'check-installandlogin:dual,noreturn']);
33 $routes->post('/dashboard/widget_positions', 'Dashboard::widget_positions');
34 // Authentication
35 $routes->add('/install', 'Install::index');
36 $routes->add('/login', 'Login::index');
37 $routes->post('login/register', 'Login::register');
38 $routes->get('login/lost', 'Login::lost');
39 $routes->post('login/lost', 'Login::lost');
40 $routes->get('/reset/:any', 'Login::reset/$1');
41 $routes->post('login/reset/:any', 'Login::reset/$1');
42 $routes->get('/register', 'Login::register');
43 $routes->post('/register', 'Login::register');
44 $routes->get('/settings', 'Settings::index', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
45 $routes->post('/settings', 'Settings::index', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
46 $routes->add('/profile/:num', 'Profile::index/$1', ['filter' => 'check-login:dual,noreturn']);
47 $routes->get('/logout', 'Login::logout');
48 $routes->add('/settings/update', 'Settings::update');
49 // Users
50 $routes->get('/profile', 'Profile::index');
51 $routes->get('/users', 'Users::index', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
52 $routes->get('/users/add', 'Users::add', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
53 $routes->post('/users/add', 'Users::add', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
54 $routes->post('/users/delete_submissions', 'Users::delete_submissions', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
55 $routes->get('/users/list_excel', 'Users::list_excel', ['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
56 // Notifications and scoreboard
57 $routes->get('/notifications', 'Notifications::index', ['filter' => 'check-login:dual,noreturn']);
58 $routes->get('/notifications/add', 'Notifications::add', ['filter' => 'check-login:dual,noreturn']);
59 $routes->post('/notifications/add', 'Notifications::add', ['filter' => 'check-login:dual,noreturn']);
60 $routes->add('/notifications/edit/:num', 'Notifications::edit/$1', ['filter' => 'check-login:dual,noreturn']);
61 $routes->post('/notifications/delete', 'Notifications::delete', ['filter' => 'check-login:dual,noreturn']);
62 $routes->post('/notifications/check', 'Notifications::check', ['filter' => 'check-login:dual,noreturn']);
63 $routes->get('/scoreboard', 'Scoreboard::index', ['filter' => 'check-loginandcli:dual,noreturn']);
64 $routes->add('/server_time', 'Server_time::index', ['filter' => 'check-loginandisajax:dual,noreturn']);
65 // Logs, hof, moss, rejudge, queue
66 $routes->get('logs', 'Logs::index', ['filter' => 'check-login:dual,noreturn']);
67 $routes->get('halloffame', 'Halloffame::index', ['filter' => 'check-login:dual,noreturn']);
68 $routes->get('moss/:num', 'Moss::index/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);
69 $routes->post('moss/:num', 'Moss::index/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);
70 $routes->get('moss/update/:num', 'Moss::update/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);
71 $routes->post('moss/update/:num', 'Moss::update/$1', ['filter' => 'check-loginandlevelHead:dual,noreturn']);

```

```

52 $routes->get('queue', 'Queue::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
53 $routes->post('queue', 'Queue::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
54 $routes->post('queue/resume', 'Queue::resume', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
55 $routes->post('queue/pause', 'Queue::pause', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
56 $routes->post('queue/empty_queue', 'Queue::empty_queue', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
57 $routes->cli('queueprocess/run', 'Queueprocess::run');
58 $routes->get('rejudge', 'Rejudge::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
59 $routes->post('rejudge', 'Rejudge::index', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
60 $routes->get('rejudge/:num', 'Rejudge::index/$1', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
61 $routes->post('rejudge/rejudge_single', 'Rejudge::rejudge_single', ['filter' => 'check-loginAndLevelHead:dual,noreturn']);
62 //Assignments
63 $routes->get('/assignments', 'Assignments::index', ['filter' => 'check-login:dual,noreturn']);
64 $routes->get('/assignments/add', 'Assignments::add', ['filter' => 'check-login:dual,noreturn']);
65 $routes->post('/assignments/add', 'Assignments::add', ['filter' => 'check-login:dual,noreturn']);
66 $routes->post('/assignments/select', 'Assignments::select', ['filter' => 'check-login:dual,noreturn']);
67 $routes->get('/assignments/edit/:num', 'Assignments::edit/$1', ['filter' => 'check-login:dual,noreturn']);
68 $routes->post('/assignments/edit/:num', 'Assignments::edit/$1', ['filter' => 'check-login:dual,noreturn']);
69 $routes->get('/assignments/delete/:num', 'Assignments::delete/$1', ['filter' => 'check-login:dual,noreturn']);
70 $routes->post('/assignments/delete/:num', 'Assignments::delete/$1', ['filter' => 'check-login:dual,noreturn']);
71 $routes->get('/assignments/pdf/:any/(:any)', 'Assignments::pdf/$1/$2/$3', ['filter' => 'check-login:dual,noreturn']);
72 $routes->get('/assignments/pdf/:num', 'Assignments::pdf/$1', ['filter' => 'check-login:dual,noreturn']);
73 $routes->post('/assignments/pdfCheck/:num', 'Assignments::pdfCheck/$1');
74 $routes->get('/assignments/pdfCheck/:num', 'Assignments::pdfCheck/$1');
75 $routes->get('assignments/downloadtestsdesc/:num', 'Assignments::downloadtestsdesc/$1');
76 $routes->get('assignments/download_submissions/:any/(:num)', 'Assignments::download_submissions/$1/$2');
77 // Submit and submissions
78 $routes->get('submit', 'Submit::index', ['filter' => 'check-login:dual,noreturn']);
79 $routes->post('submit', 'Submit::index', ['filter' => 'check-login:dual,noreturn']);
80 $routes->get('submit/load/:num', 'Submit::load/$1', ['filter' => 'check-login:dual,noreturn']);
81 $routes->post('submit/save/:any', 'Submit::save/$1');
82 $routes->post('submit/save', 'Submit::save');
83 $routes->get('submit/get_output/:num', 'Submit::get_output/$1');
84 $routes->get('submissions/all/user/:any', 'Submissions::all');
85 $routes->get('submissions/final', 'Submissions::the_final', ['filter' => 'check-login:dual,noreturn']);
86 $routes->get('submissions/final/:any', 'Submissions::the_final/$1', ['filter' => 'check-login:dual,noreturn']);
87 $routes->post('submissions/view_code', 'Submissions::view_code');
88 $routes->post('submissions/select', 'Submissions::select');
89 $routes->get('submissions/all_excel', 'Submissions::all_excel');
90 $routes->get('submissions/final_excel', 'Submissions::final_excel');
91 $routes->get('submissions/download_file/:any/(:num)/(:num)', 'Submissions::download_file/$1/$2/$3/$4');
92 $routes->get('submissions/all', 'Submissions::all');
93 // Problems
94 $routes->get('/problems', 'Problems::index', ['filter' => 'check-login:dual,noreturn']);
95 $routes->get('/problems/:num', 'Problems::index/$1');
96 $routes->get('/problems/:num/(:num)', 'Problems::index/$1/$2');
97 $routes->get('/problems/edit/:any/(:num)/(:num)', 'Problems::edit/$1/$2/$3');
98 $routes->post('/problems/edit/:any/(:num)/(:num)', 'Problems::edit/$1/$2/$3');

```

- 49 Kode 4.7 merupakan seluruh *routes* yang didefinisikan secara explisit sesuai dengan kegunaannya.
50 Terdapat juga penambahan *filters* untuk melakukan pengecekan sebelum *controller* diinisiasi.
51 Pembentukan *routes* secara manual dilakukan karena alasan keamanan dan juga konfigurasi *filters*
52 yang berbeda untuk setiap *controller*.

53 app/Config/Secrets.php

- 54 File ini merupakan file yang ditambahkan diluar dari struktur standart dari aplikasi *CodeIgniter 4*.
55 File ini berisikan seluruh kebutuhan konfigurasi yang tidak terdapat pada *config* lainnya. Kode 4.8
56 merupakan pemindahan file *Secrets.php* menuju *app/Config*.

Kode 4.8: Penambahan file *secrets*

```

57
58 1 namespace Config;
59 2
60 3 use CodeIgniter\Config\BaseConfig;
61 4 class Secrets extends BaseConfig
62 5 {
63 6     public $shj_authenticate = 'radius';
64 7
65 8     public $shj_radius = [
66 9         "server" => "localhost",
67 0         "secret" => "i-have-no-secret"
68 1     ];
69 2

```

```

113 // LDAP Settings refers to
114 // @link https://adldap2.github.io/Adldap2/#/setup?id=options
115 public $shj_ldap = [
116     "hosts" => ["dc3.ftis.unpar"],
117     "base_dn" => "DC=FTIS,DC=UNPAR",
118     "username"=> "",
119     "password"=> "",
120     "account_suffix"    => "@ftis.unpar"
121 ];
122
123 public $shj_mail = [
124     'protocol' => 'smtp',
125     'SMTPHost' => 'ssl://smtp.mailgun.org',
126     'SMTPPort' => 465,
127     'SMTPUser' => '',
128     'SMTPPass' => '',
129     'mailType'  => 'html',
130     'charset'   => 'utf-8'
131 ];
132

```

Kode 4.8 merupakan pemindahan kode menuju app/Config. Terdapat penghapusan sintaks `defined('BASEPATH') OR exit('No direct script access allowed');` pada baris pertama kalimat. Sintaks yang telah dihapus akan digantikan oleh sintaks *class* seperti pada baris pertama dan nama kelas akan *extend BaseConfig*. Terdapat juga penambahan sintaks *namespace* dan *use*. File ini akan digunakan dan dipanggil pada *controller* untuk melakukan autentikasi pengguna.

28 app/Config/Security.php

File ini akan berisikan seluruh konfigurasi yang dipindahkan dari application/config/config.php. Kode 4.9 merupakan pemindahan data token *csrf* dan *cookie* menuju Security.php.

Kode 4.9: Pemindahan file config menuju Security

```

31 public string $tokenName = 'shj_csrf_token';
32 public string $cookieName = 'shjcsrfToken';

```

Kode 4.9 merupakan pemindahan sintaks menuju Security.php. Sintaks pada baris pertama merupakan nama *token* yang akan digunakan sedangkan sintaks kedua merupakan nama *cookie* yang akan digunakan. Sintaks pada CodeIgniter 4 tidak akan menggunakan *array* namun akan disimpan menuju variabel. Sintaks lainnya tidak akan diubah karena akan menggunakan konfigurasi *default* dari CodeIgniter 4.

40 app/Config/Session.php

File ini akan berisikan seluruh konfigurasi yang dipindahkan dari application/config/config.php. Kode 4.10 merupakan pemindahan data konfigurasi *session* menuju Session.php.

Kode 4.10: Pemindahan file config menuju Session

```

43 public string $driver = 'CodeIgniter\Session\Handlers\DatabaseHandler';
44 public string $cookieName = 'shjsession';

```

Kode 4.10 merupakan pemindahan data menuju Session.php. Data yang dipindahkan berupa *cookieName* yang akan disimpan menuju variabel. Selanjutnya akan ada perubahan *driver* dimana akan menggunakan sintaks diatas karena akan disimpan menuju *database*.

1 app/Config/Validation.php

- 2 File ini berisikan aturan yang dibentuk secara manual oleh pengguna untuk melakukan validasi
3 sebelum *request* diproses oleh aplikasi. Kode [4.11](#) merupakan aturan validasi data yang dibentuk
4 secara manual oleh pengguna.

Kode 4.11: Perancangan aturan yang dibentuk secara manual pada file Validation.php

```
5  class MyRules
6  {
7      public function password_check($str): bool
8      {
9          if (strlen($str) == 0 OR (strlen($str) >= 6 && strlen($str) <= 200))
10         return TRUE;
11     return FALSE;
12 }
13
14
15  public function password_again_check($str) :bool
16 {
17     $request = \Config\Services::request();
18     if ($request->getPost('password') !== $request->getPost('password_again'))
19         return FALSE;
20     return TRUE;
21 }
22
23
24  public function role_check($role)
25 {
26     $user = new User();
27     if ($user->level <= 2)
28         return ($role == '');
29
30     // Admins can change everybody's user role:
31     $roles = array('admin', 'head_instructor', 'instructor', 'student');
32     return in_array($role, $roles);
33 }
34
35 /**
36 * Checks whether a user with this email exists
37 */
38 public function email_check($email,$edit_username):bool
39 {
40     $edit_username = explode(',', $edit_username);
41     $user_model = new UserModel();
42     if ($user_model->have_email($email, $edit_username))
43         return FALSE;
44     return TRUE;
45 }
46
47 /**
48 * checks whether the entered registration code is correct or not
49 *
50 */
51 public function _registration_code($code){
52     $settings_model = new SettingsModel();
53     $rc = $settings_model->get_setting('registration_code');
54     if ($rc == '0')
55         return TRUE;
56     if ($rc == $code)
57         return TRUE;
58     return FALSE;
59 }
60
61 /**
62 * Required
63 *
64 * @param string
65 * @return bool
66 */
67 public function required($str)
68 {
69     return is_array($str) ? (bool) count($str) : ($str !== '');
70 }
71
72
73 /**
74 * Is Lowercase
75 */
```

```

72  *
73  * @param $str
74  * @return bool
75  */
76  public function lowercase($str)
77  {
78      return (strtolower($str) === $str);
79  }
80
81  // -----
82
83
84  public function _check_language($str)
85  {
86      if ($str=='0')
87          return FALSE;
88      if (in_array( strtolower($str),array('c', 'c++', 'python 2', 'python 3', 'java', 'zip', 'pdf', 'txt')))
89          return TRUE;
90      return FALSE;
91  }
92
93
94  // -----
95  // Used in Submissions.php
96
97  public function _check_type($type)
98  {
99      return ($type === 'code' || $type === 'result' || $type === 'log');
100 }
101 }
```

Kode 4.11 merupakan pembentukan aturan secara manual yang dipindahkan dari *controller* *Login.php*, *Profile.php*, *Submissions.php*, dan *Submit.php*. Aturan yang dibentuk secara manual dapat digunakan seperti aturan lainnya yang tersedia pada *CodeIgniter 4*. Aturan yang dibentuk terdapat beberapa penambahan seperti pemanggilan *library*, *model*, dan penambahan sintaks untuk memecah *array*.

4.2.2 Controllers

Controller terdapat perubahan pada bagian fungsi *__construct()* dimana sekarang tidak dapat mengembalikan sesuatu. Oleh karena itu, akan dibentuk beberapa *filters* 4.2.1 untuk melakukan pengecekan terhadap fungsi yang sebelumnya terdapat pada *__construct()*.

app/Controllers

Direktori ini berisikan seluruh *controller* yang dipindahkan dari *application/controllers*. Berikut merupakan isi pada direktori ini:

- *Assignments.php*
- *BaseController.php*
- *Dashboard.php*
- *Halloffame.php*
- *Install.php*
- *Login.php*
- *Logs.php*
- *Moss.php*
- *Notifications.php*
- *Problems.php*
- *Profile.php*

- 1 • Queue.php
- 2 • Queueprocess.php
- 3 • Rejudge.php
- 4 • Scoreboard.php
- 5 • Server_time.php
- 6 • Settings.php
- 7 • Submissions.php
- 8 • Submit.php
- 9 • Users.php

10 Controllers terdapat perubahan dan penambahan baik dalam kelas yang dilakukan *extends*
 11 maupun dalam pemanggilan kelas lain seperti *model* dan *helpers*. Beberapa *helpers* yang dipanggil
 12 melalui *autoload* akan dipindahkan menuju *BaseController*. Kode 4.12 merupakan perubahan yang
 13 terdapat pada *controller Logs.php*.

Kode 4.12: Perubahan kode *controller Logs.php* pada *CodeIgniter 4*

```

14 namespace App\Controllers;
15 1
16 2
17 3 use App\Controllers\BaseController;
18 4 use App\Models\AssignmentModel;
19 5 use App\Models\LogsModel;
20 6 use App\Models\User;
21 7
22 8 class Logs extends BaseController
23 9 {
24 10     protected $session;
25 11     protected $user;
26 12     protected $logs_model;
27 13     protected $assignment_model;
28 14
29 15     public function __construct()
30 16     {
31 17         $this->session = session();
32 18         $this->logs_model = new LogsModel();
33 19         $this->assignment_model = new AssignmentModel();
34 20         $this->user = new User();
35 21         if ( $this->user->level <= 2 ) // permission denied
36 22             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
37 23     }
38 24
39 25     public function index()
40 26     {
41 27
42 28         $data = array(
43 29             'logs' => $this->logs_model->get_all_logs(),
44 30             'selected' => 'logs',
45 31             'user' => $this->user,
46 32             'all_assignments' => $this->assignment_model->all_assignments(),
47 33             'finish_time' => $this->user->selected_assignment['finish_time'],
48 34             'extra_time' => $this->user->selected_assignment['extra_time'],
49 35         );
50 36
51 37         return view('pages/admin/logs', $data);
52 38     }
53 39 }
```

55 Kode 4.12 terdapat perubahan dimana sekarang akan *extends BaseController*. Terdapat juga
 56 penghapusan sintaks `defined('BASEPATH') OR exit('No direct script access allowed');`.
 57 Terdapat penambahan sintaks *namespace* dan juga beberapa sintaks untuk memanggil *models*.
 58 *Controller* juga memiliki perubahan dalam mengembalikan *view* dimana sekarang menggunakan
 59 sintaks `return view`. Selain itu terdapat penambahan pada *BaseController.php* untuk melakukan
 60 inisiasi terhadap *helpers* dan juga beberapa *library* yang akan digunakan. Kode 4.13 merupakan
 61 penambahan *helpers* yang dapat diakses oleh seluruh *controller*

Kode 4.13: Penambahan sintaks pada BaseController

```

1 protected $helpers = ['text','url','shj_helper','form','cookie','string','filesystem'];
2 /**
3  * Be sure to declare properties for any property fetch you initialized.
4  * The creation of dynamic property is deprecated in PHP 8.2.
5  */
6 protected $parsedown;
7 protected $twig;
8
9 /**
10 * Constructor.
11 */
12 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
13 {
14     // Do Not Edit This Line
15     parent::initController($request, $response, $logger);
16
17     // Preload any models, libraries, etc, here.
18     $this->parsedown = new Parsedown();
19     $this->twig = new Twig();
20
21 }

```

Kode 4.13 merupakan pemindahan dan penambahan *helpers* dari *autoload* menuju *BaseController*. *Helpers* akan ditambahkan menuju *array \$helpers* sedangkan beberapa *thirdparty library* akan dilakukan inisiasi di dalam fungsi *initController*. Seluruh *helpers* yang telah dipindahkan dapat diakses pada seluruh *controller*. Penambahan ini dilakukan untuk mempermudah dalam penggunaan *helpers* pada seluruh *controller* agar tidak perlu diinisiasi lagi pada setiap *controller* yang membutuhkan.

30 IncomingRequest

31 *Input* digantikan oleh fungsi *request* dengan perubahan cara pemanggilan sintaks. *Validation* hanya
32 akan diinisiasikan pada fungsi *__construct* setiap *model* yang membutuhkan. Fungsi tidak akan
33 diinisiasikan pada *controller* karena *CodeIgniter 4* sudah menyediakan fitur ini pada *controller*
34 sehingga hanya perlu dilakukan pemanggilan. Kode 4.14 merupakan inisiasi *request* pada *model*.

Kode 4.14: Perancangan inisiasi *request* pada *__construct*

```

35
36 protected $request;
37
38
39 public function __construct()
40 {
41     $this->request = \Config\Services::request();
42 }
43

```

44 Kode 4.14 merupakan inisiasi yang dilakukan pada *model*. *Controller* tidak perlu melakukan
45 inisiasi terhadap fungsi ini karena sudah terinisiasi pada *BaseController*. Pengambilan *input* yang
46 dilakukan oleh pengguna dilakukan menggunakan sintaks pada kode 4.15.

Kode 4.15: Perancangan penggunaan *request*

```

47
48 if($this->request->isAJAX())
49 $this->request->getPost('assignment_select')
50

```

51 Kode 4.15 merupakan cara penggunaan *request* untuk melakukan pengecekan dan pengambilan
52 data. Sintaks berubah dari yang sebelumnya menggunakan *snakecase* menjadi *camelCase*.

53 4.2.3 Filters

54 Pada *CodeIgniter 4* *__construct()* tidak dapat mengembalikan sesuatu oleh karena itu akan
55 dibentuk beberapa *filters* untuk melakukan pengecekan. Beberapa *filters* yang dibentuk antara lain

1 berfungsi untuk mengecek apakah dijalankan dari *command line interface*, apakah sudah *install* dan
 2 apakah sudah *login*, apakah sudah *login* dan dijalankan dari *command line interface*, apakah
 3 sudah *login* dan apakah *request* berupa *AJAX*, dan apakah sudah *login* dan pengecekan terhadap
 4 *role* pengguna. Terdapat total tujuh buah *filters* yang akan dibentuk. Kode 4.16 merupakan sintaks
 5 untuk melakukan pengecekan apakah dijalankan dari *command line interface*.

Kode 4.16: Perancangan kode pada *Filters CheckCLI.php*

```

6  public function before(RequestInterface $request, $arguments = null)
7  {
8      $request = \Config\Services::request();
9
10     if ($request->isCLI())
11         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
12
13 }
14
15 }
```

15 Kode 4.16 merupakan pemindahan kode dari *__construct* menuju *filters CheckCLI*. *__construct*
 16 terdapat pada *controller* dimana fungsi diatas digunakan pada *controller Queueprocessphp*. Kode
 17 ini akan mengecek apakah *request* yang diberikan oleh pengguna merupakan *command line interface*.
 18 Apabila *request* yang diberikan bukan berupa itu maka akan diberikan error berupa halaman tidak
 19 ditemukan. Kode 4.17 merupakan *filters* yang dibentuk untuk melakukan pengecekan apakah
 20 pengguna sudah melakukan *install* dan *login*.

Kode 4.17: Perancangan kode pada *Filters CheckInstallAndLogin.php*

```

21  public function before(RequestInterface $request, $arguments = null)
22  {
23      $db = Database::connect();
24
25
26      if ( !$db->tableExists('sessions'))
27          return redirect()->to('/install');
28
29
30      $session = \Config\Services::session();
31      if ( !$session->get('logged_in')) // if not logged in
32          return redirect()->to('/login');
33
34 }
```

34 Kode 4.17 merupakan pemindahan dari *__construct* pada *controller Dashboard*. Fungsi pada
 35 *filters* ini akan melakukan pengecekan apakah sudah terdapat tabel *session* dan apakah pengguna
 36 sudah melakukan *login*. Selanjutnya *filter* akan mengembalikan pengguna kepada *routes* yang
 37 sesuai apabila tidak sesuai dengan kondisinya. Kode 4.18 merupakan *filters* yang dibentuk untuk
 38 melakukan pengecekan apakah pengguna sudah melakukan *login*.

Kode 4.18: Perancangan kode pada *Filters CheckLogin.php*

```

39  public function before(RequestInterface $request, $arguments = null)
40  {
41      $session = \Config\Services::session();
42
43
44      if ( !$session->get('logged_in')) // if not logged in
45          return redirect()->to('login');
46
47 }
```

48 Kode 4.18 merupakan pemindahan dari *__construct* pada *controller Profile, Notifications,*
 49 *Logs, Assignments, Submit, Submission, dan Problems*. Fungsi pada *filters* ini akan melakukan
 50 pengecekan apakah pengguna sudah melakukan *login*. Selanjutnya *filter* akan mengembalikan
 51 pengguna kepada *routes* yang sesuai. Kode 4.19 merupakan *filters* yang dibentuk untuk melakukan
 52 pengecekan apakah pengguna menjalankan *request* dari *command line interface* dan apakah
 53 pengguna sudah melakukan *login*.

Kode 4.19: Perancangan kode pada *Filters CheckLoginandCLI.php*

```
1 public function before(RequestInterface $request, $arguments = null)
2 {
3     $session = \Config\Services::session();
4     $request = \Config\Services::request();
5
6     if ($request->isCLI())
7         return;
8
9     if ( !$session->get('logged_in') ) // if not logged in
10        return redirect()->to('login');
11
12 }
```

Kode 4.19 merupakan pemindahan dari `__construct` pada *controller Scoreboard*. Fungsi pada *filters* ini akan melakukan pengecekan apakah *controller* dijalankan melalui *command line* dan apakah pengguna sudah *login*. Selanjutnya *filter* akan mengembalikan pengguna kepada *routes* yang sesuai apabila tidak sesuai dengan kondisinya. Kode 4.20 merupakan *filters* yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan login dan apakah *request* yang diberikan pengguna berupa *ajax*.

Kode 4.20: Pemindahan kode pada *Filters CheckLoginandisAjax.php*

```
19
20     public function before(RequestInterface $request, $arguments = null)
21 {
22     $session = \Config\Services::session();
23     $request = \Config\Services::request();
24
25     if ( !$session->get('logged_in') ) // if not logged in
26         return redirect()->to('/login');
27     if ( !$request->isAJAX() )
28         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
29 }
30
```

Kode 4.20 merupakan pemindahan dari `__construct` pada `controller Server_time`. Fungsi pada `filters` ini akan melakukan pengecekan apakah pengguna sudah `login` dan apakah `request` berupa `ajax`. Selanjutnya `filter` akan mengembalikan pengguna kepada `routes` yang sesuai apabila tidak sesuai dengan kondisinya. Kode 4.21 merupakan `filters` yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan login dan apakah `role` dari pengguna dapat mengakses situs tersebut.

Kode 4.21: Pemindahan kode pada *Filters CheckLoginandLevelAdmin.php*

```
37
38 1 public function before(RequestInterface $request, $arguments = null)
39 2 {
40 3     $session = \Config\Services::session();
41 4     $user = new User();
42 5
43 6     if ( !$session->get('logged_in') ) // if not logged in
44 7         return redirect()->to('/login');
45 8     if ( $user->level <= 2 ) // permission denied
46 9         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
47 0 }
```

Kode 4.21 merupakan pemindahan dari `__construct` pada `controller Server_time`. Fungsi pada `filters` ini akan melakukan pengecekan terhadap `level` pengguna apakah `admin` dan apakah pengguna sudah melakukan `login`. Selanjutnya `filter` akan mengembalikan pengguna kepada `routes` yang sesuai apabila tidak sesuai dengan kondisinya. Kode 4.22 merupakan `filters` yang dibentuk untuk melakukan pengecekan apakah pengguna sudah melakukan login dan apakah `role` dari pengguna dapat mengakses situs tersebut.

Kode 4.22: Pemindahan kode pada *Filters CheckLoginandLevelHead.php*

```
55     public function before(RequestInterface $request, $arguments = null)
56     {
57         $this->before($request, $arguments);
58     }
59 }
```

```

13     $session = \Config\Services::session();
14     $user = new User();
15
16     if ( !$session->get('logged_in')) // if not logged in
17         return redirect()->to('/login');
18     if ( $user->level <= 1) // permission denied
19         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
20 }

```

10 Kode 4.21 merupakan pemindahan dari `__construct` pada *controller Server_time*. Fungsi
 11 pada *filters* ini akan melakukan pengecekan *level* pengguna dan apakah pengguna sudah melakukan
 12 *login*. Selanjutnya *filter* ini akan mengembalikan pengguna kepada *routes* yang sesuai apabila
 13 tidak sesuai dengan kondisinya. Setelah dibentuk *filters*, selanjutnya akan ditambahkan menuju
 14 file *Filters.php* untuk mendefinisikan nama yang selanjutnya akan dimasukkan menuju *routes*.
 15 Penambahan menuju file *Filters.php* dapat dilihat pada subbab 4.2.1. Setelah ditambahkan, *filters*
 16 akan dipanggil pada *routes* yang membutuhkan pengecekan. Kode 4.23 merupakan penambahan
 17 *filters* pada *routes* sesuai dengan kebutuhannya.

Kode 4.23: Penambahan *filter* pada *routes*

```

18 $routes->get('/settings','Settings::index',['filter' => 'check-loginandlevelAdmin:dual,noreturn']);
20

```

21 Kode 4.23 menambahkan *filter* yang sudah dibentuk dan dinamakan setelah penulisan nama
 22 *controller* dan fungsinya.

23 4.2.4 *Helpers*

24 Direktori ini berisikan *helpers* yang dibentuk secara manual oleh pengguna bernama *shj_helper*.
 25 *Helpers* terdapat perubahan dan penghapusan sintaks. Sintaks `defined('BASEPATH') OR exit('No`
 26 `direct script access allowed');` akan dihapus dan akan ditambahkan beberapa *model* yang
 27 digunakan pada *helper* ini. *Model* yang ditambahkan berupa *settings_model*.

28 4.2.5 *Libraries*

29 *Libraries* terdapat beberapa perubahan dan penghapusan fungsi sehingga akan digantikan. Berikut
 30 merupakan perancangan perubahan fungsi pada *CodeIgniter 4*.

31 *Emails*

32 *Emails* pada *CodeIgnier 4* terdapat perubahan sintaks dan cara pemanggilan sehingga akan dipindahkannya sesuai dengan sintaks yang baru. Sintaks berubah dari yang sebelumnya menggunakan *snakecase* menjadi menggunakan *camelcase*. Kode 4.24 merupakan contoh penggunaan *library email*.

Kode 4.24: Perubahan penggunaan sintaks pada *library emails*

```

35 $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
36 $this->email->setTo($user[1]);
37 $this->email->setSubject('SharIF Judge Username and Password');
38 $text = $this->settings_model->get_setting('add_user_mail');
39 $text = str_replace('{SITE_URL}', base_url(), $text);
40 $text = str_replace('{ROLE}', $user[4], $text);
41 $text = str_replace('{USERNAME}', $user[0], $text);
42 $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
43 $text = str_replace('{LOGIN_URL}', base_url(), $text);
44 $this->email->setMessage($text);
45 $this->email->send()
46

```

48 Kode 4.24 memiliki sintaks dengan nama sama namun terdapat perubahan menjadi *camelcase*.

1 **Working with Uploaded Files**

2 *Working with uploaded files* terdapat perubahan pada beberapa sintaks dan validasi terhadap *file*
 3 yang telah diunggah. Konversi aplikasi *SharIF Judge* akan menggunakan fungsi ini dengan beberapa
 4 perubahan sintaks sesuai dengan dokumentasinya. Kode 4.25 merupakan perubahan yang terdapat
 5 pada *library* ini.

Kode 4.25: Perancangan perubahan *library upload* pada *CodeIgniter 4*

```

6 1 $zip_uploaded = $this->request->getFile('tests_desc');
6 2   if ( $_FILES['tests_desc']['error'] === UPLOAD_ERR_NO_FILE ){
6 3     $this->messages[] = array(
6 4       'type' => 'notice',
6 5       'text' => "Notice: You did not upload any zip file for tests. If needed, upload by editing assignment."
6 6     );
6 7   }
6 8   elseif ( $zip_uploaded->getExtension() != "zip" ){
6 9     $this->messages[] = array(
6 10       'type' => 'error',
6 11       'text' => "Error: Error uploading tests zip file: The filetype you are attempting to upload is not allowed."
6 12     );
6 13   }
6 14   else{
6 15     $zip_uploaded->move($assignments_root);
6 16     $this->messages[] = array(
6 17       'type' => 'success',
6 18       'text' => "Tests (zip file) uploaded successfully."
6 19     );
6 20   }
6 21 }
```

28 Kode 4.25 merupakan perubahan yang terdapat pada *library upload*. Pengambilan file digantikan
 29 dengan sintaks `$this->request->getFile('tests_desc')` dengan parameter berupa nama dari
 30 *tag form* yang sudah dibentuk. Selanjutnya dilakukan pengecekan terhadap file yang sudah di
 31 unggah dan memberikan beberapa *error message* sesuai dengan kondisinya. File yang sudah di
 32 unggah dipindahkan menuju direktori yang disimpan pada variabel `assignments_root`.

33 Konfigurasi aturan terhadap *upload* sudah tidak terdapat pada *CodeIgniter 4* sehingga akan
 34 diubah dan digantikan pada kondisi di baris kedua. Kondisi tersebut dibentuk untuk melakukan
 35 pengecekan tipe file yang sudah diunggah dan memberikan *error*. *Error* yang dihasilkan juga
 36 sudah tidak dinamik karena tidak menggunakan *validation* sehingga disesuaikan dengan *error* yang
 37 terdapat pada *CodeIgniter 4*.

38 **Working with URIs**

39 Kelas *URI* digantikan oleh fungsi *working with URIs* dengan perubahan dan penambahan fungsi.
 40 *Library* ini akan diinisiasi pada fungsi `__construct controller Submissions.php`. Kode 4.26
 41 merupakan inisiasi *library* ini dan terdapat penambahan sintaks karena terdapat fungsi yang dihapus.

Kode 4.26: Perancangan inisiasi dan penambahan sintaks *library URIs* pada `__construct`

```

42 $temp_inp = $this->uri->getSegments();
43 $input=[];
44 foreach($temp_inp as $key => $value){
45   if($key != (count($temp_inp)-1) && ($key%2 != 0)){
46     $input[$value] = $temp_inp[$key+1];
47   }else{
48     break;
49   }
50 }
```

53 Kode 4.26 merupakan sintaks untuk mengambil *segment* yang terdapat pada *URI*. *Segment* ini akan
 54 disimpan dalam sebuah *array* dan tidak membentuk *associative array*. Oleh karena itu, terdapat
 55 penambahan sintaks `foreach` untuk mengubah *array* tersebut menjadi sebuah *associative array*.

1 **Session**

- 2 Sintaks penggunaan *session* akan berubah dan *session* akan dilakukan inisiasi sebelum pemakaian.
- 3 Berikut merupakan perubahan dalam melakukan inisiasi pada *session*.

Kode 4.27: Perancangan inisiasi *library session*

```

4 51
6 2 protected $session;
7 3
8 4 public function selected_assignment($username){
9 5     $this->session = session();
10 6 }
11 7 }
```

- 13 Kode 4.27 merupakan sintaks untuk melakukan inisiasi terhadap *session*. *Session* akan diinisiasikan pada setiap file yang menggunakan *library* ini. Penggunaan *session* terdapat perubahan dari sintaks untuk pengambilan data, membersihkan data, dan penetapan data. Berikut merupakan perubahan dari sintaks pada *library session*:

Kode 4.28: Perancangan penggunaan *library session*

```

17
18 1 $this->session->get('logged_in')
19 2 $this->session->set($login_data);
20 3 $this->session->destroy();
```

- 22 Kode 4.28 merupakan cara penggunaan *library session*. Sintaks pada baris pertama berfungsi untuk mengambil data dengan nama *logged_in*. Sintaks pada baris kedua berfungsi untuk menetapkan sebuah data *session* pada variabel *login_data*. Terakhir pada baris ketiga merupakan sintaks untuk membersihkan data pada *session* pengguna tersebut.

26 **Validation**

- 27 *Form_validation* digantikan oleh fungsi *validation* dengan perubahan dan pengapusan beberapa fungsi. *Validation* akan diinisiasikan pada fungsi *__construct* setiap *controller* dan *model* yang membutuhkan. Kode 4.29 merupakan inisiasi *validation* pada *controller*.

Kode 4.29: Perancangan inisiasi *validation* pada *__construct*

```

30
31 1
32 2 protected $validation;
33 3
34 4     public function __construct()
35 5     {
36 6         $this->validation = \Config\Services::validation();
37 7     }
```

- 39 Kode 4.29 merupakan sintaks untuk melakukan inisiasi terhadap *validation*. Setiap *controller* yang menggunakan *validation* akan dideklarasikan sebuah variabel bernama *validation* agar dapat dipanggil pada seluruh fungsi yang membutuhkan pada kelas tersebut. *Validation* dilanjutkan dengan penetapan aturan untuk *tag form* yang diinginkan. Berikut merupakan contoh penetapan aturan untuk mengumpulkan sebuah data pada *form* dengan nama *username*.

Kode 4.30: Perancangan perubahan konfigurasi aturan pada *library validation*

```

44
45 1 $this->validation->setRule('username', 'username', 'required|min\length[3]|max\length[20]);
```

- 47 Kode 4.30 akan menetapkan aturan terhadap *input* yang akan masukan oleh pengguna sesuai dengan nama *formnya*. Sedangkan sintaks untuk penetapan aturan berubah menggunakan *camelCase*. Beberapa aturan yang dibentuk secara manual akan dipindahkan menuju file *Validation.php*

1 dan terdapat penghapusan aturan yang sudah tidak terdapat pada *CodeIgniter 4*. Pembentukan
 2 aturan dapat dilihat pada sub bab **4.2.1** kode **4.11**. Aturan yang sudah dibentuk dapat digunakan
 3 seperti aturan lainnya dengan cara menulis nama kelasnya. Setelah aturan ditetapkan, *validation*
 4 akan dieksekusi berdasarkan *request* dari pengguna dan dilakukan validasi. Kode **4.31** merupakan
 5 perubahan penggunaan *validation* terhadap data yang sudah diberikan oleh pengguna.

Kode 4.31: Perancangan perubahan penggunaan *validation* pada *CodeIgniter 4*

```

6  if ($this->validation->withRequest($this->request)->run())
7  {
8      if ( !$this->request->isAJAX() ){
9          exit;
10     }else{
11         list($ok, $error) = $this->user_model->add_users(
12             $this->request->getPost('new_users'),
13             $this->request->getPost('send_mail'),
14             $this->request->getPost('delay')
15         );
16         return view('pages/admin/add_user_result', array('ok' => $ok, 'error' => $error));
17     }
18 }
19 }
```

21 Kode **4.31** akan menjalankan *validation* berdasarkan *request* dari pengguna. *Validation* akan
 22 tetap menggunakan sintaks **run()** namun akan ada penambahan sintaks **withRequest** dimana
 23 validasi akan dijalankan setiap ada *HTTP Request* dari pengguna. Namun, *CodeIgniter 4* tidak
 24 menyediakan fungsi **form_error** sehingga akan diubah dengan menggunakan fungsi baru bernama
 25 **validation_errors()**. Fungsi tersebut dapat digunakan untuk mengembalikan *error* apabila
 26 terdapat data yang tidak sesuai dengan aturan. *Error* tersebut dapat ditampilkan pada halaman
 27 *view* menggunakan sintaks berikut.

```
28 <?= $validation->hasError('username') ? $validation->getError('username') : '' ?>
```

29 Sintaks diatas akan melakukan pengecekan apakah terdapat *error* dari *form* bernama **username**
 30 dan apabila terdapat maka akan dikembalikan data *error* yang berasal dari *validation*. Variabel
 31 **validation** akan dikirimkan dari *controller* berisikan *library validation*.

32 Zip Archive

33 *Zip Encoding* akan digantikan dengan fungsi PHP *zip archive* karena sudah tidak tersedia pada
 34 *CodeIgniter 4*. Fungsi *zip archive* terdapat beberapa perbedaan sehingga akan disesuaikan dengan
 35 fungsi-fungsi yang ada. Kode **4.32** merupakan perubahan yang terdapat pada *zip encoding*.

Kode 4.32: Perancangan perubahan *zip encoding* menjadi *zip archive*

```

36
37 1 $this->zip = new \ZipArchive();
38 2     $this->zip->open($zipname, ZipArchive::CREATE);
39 3     for ($i=1 ; $i<=$number_of_problems ; $i++)
40 4     {
41 5
42 6         $path = "$root_path/p{$i}/in";
43 7         $options = ['add_path' => "p{$i}/in/", 'remove_all_path' => TRUE];
44 8         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
45 9
46 0         $path = "$root_path/p{$i}/out";
47 1         $options = ['add_path' => "p{$i}/out/", 'remove_all_path' => TRUE];
48 2         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
49 3
50 4         $path = "$root_path/p{$i}/tester.cpp";
51 5         if (file_exists($path))
52 6             $this->zip->addFile($path,"p{$i}/tester.cpp");
53 7
54 8         $pdf_files = glob("$root_path/p{$i}/*.pdf");
```

```
19         if ($pdf_files)
20         {
21             $path = $pdf_files[0];
22             $this->zip->addFile($path,"p{$i}/".shj_basename($path));
23         }
24
25         $path = "$root_path/p{$i}/desc.html";
26         if (file_exists($path))
27             $this->zip->addFile($path,"p{$i}/desc.html");
28
29         $path = "$root_path/p{$i}/desc.md";
30         if (file_exists($path))
31             $this->zip->addFile($path,"p{$i}/desc.md");
32     }
33
34     $pdf_files = glob("$root_path/*.pdf");
35     if ($pdf_files)
36     {
37         $path = $pdf_files[0];
38         $this->zip->addFile($path,shj_basename($path));
39     }
40     $this->zip->close();
41
42     header('Content-Type: application/zip');
43     header('Content-disposition: attachment; filename=' . $zipname);
44     header('Content-Length: ' . filesize($zipname));
45     readfile($zipname);
```

Kode 4.32 merupakan perubahan dari *zip encoding* menjadi *zip archive*. *Zip archive* akan dilakukan inisiasi pada variabel `zip`. Selanjutnya akan dibuka file `zip` menggunakan sintaks `open` yang menerima dua buah parameter. Parameter pertama berisikan nama `zip` file yang ingin dibentuk sedangkan parameter kedua berisikan mode `zip` yang diinginkan. Sintaks `addGlob` akan memasukan seluruh file sesuai dengan pattern yang telah ditentukan. Sintaks `addFile` akan memasukan file sesuai dengan `path` yang telah ditentukan. Selanjutnya `zip` akan ditutup menggunakan sintaks `close` dan dapat diunduh oleh pengguna menggunakan fungsi `header` yang disediakan oleh PHP.

Selain untuk membentuk file *zip*, *zip archive* juga mendukung fungsi untuk melakukan *unzip* sehingga akan menggantikan *library unzip* yang sebelumnya dibentuk oleh Phil Sturgeon. Kode 4.33 merupakan penggunaan *zip archive* untuk *unzip* sebuah file pada *controller*.

Kode 4.33: Perancangan perubahan *unzip* menggunakan *zip archive* pada *controller*

```
39
40 1 $this->unzip = new ZipArchive();
41 2     // Create a temp directory
42 3     $tmp_dir_name = "shj_tmp_directory";
43 4     $tmp_dir = "$assignments_root/$tmp_dir_name";
44 5
45 6     // Extract new test cases and descriptions in temp directory
46 7     $this->unzip->open("$assignments_root/".$zip_uploaded->getName());
47 8     $extract_result = $this->unzip->extractTo($tmp_dir);
48 9     $this->messages[] = array(
49 0         'type' => 'error',
50 1         'text' => " Zip Extraction Error: ".$this->unzip->getStatusString(),
51 2     );
52
```

Kode 4.33 merupakan perubahan melakukan *unzip* menggunakan *zip archive*. Sintaks pada baris pertama akan melakukan inisiasi terhadap *zip archive* sedangkan baris kedua dan ketiga akan membentuk variabel dengan nama direktori yang dituju. Sintaks *open* akan membuka sebuah file *zip* sesuai dengan nama *zip* tersebut. Sintaks selanjutnya akan melakukan *extract* file *zip* tersebut menuju direktori yang telah dibentuk dan menyimpannya pada sebuah variabel. Selanjutnya pengguna dapat mengambil status dari hasil *extract* yang telah dilakukan.

1 ***Password_hash***

2 *Library* ini tidak akan digunakan dan akan digantikan oleh *password hash* yang disediakan oleh
 3 PHP. *Library Password_hash* merekomendasikan pengguna untuk menggunakan fungsi *native* yang
 4 disediakan oleh PHP apabila aplikasi mendukung PHP versi 5.5 ke atas. Sehingga, akan dilakukan
 5 konversi menggunakan fungsi yang disediakan oleh PHP bernama `password_hash()`. Seluruh
 6 penggunaan *library* ini akan diubah menggunakan fungsi yang disediakan oleh PHP dengan metode
 7 *hashing* sama yaitu *CRYPT_BLOWFISH*. Perubahan fungsi *hashing* ini bersifat *backward compatible*
 8 sehingga dapat menggunakan *database* aplikasi terdahulu tanpa perlu membentuk data baru. Berikut
 9 merupakan contoh pengubahan kode dari *phpass* menjadi *password_hash*.

10 `'password' => $this->password_hash->HashPassword($password)`

11 menjadi

12 `'password' => password_hash($password,PASSWORD_BCRYPT)`

13 Sintaks `password_hash()` diatas menerima dua buah parameter yakni data yang ingin dienkripsi
 14 dan tipe enkripsi. Enkripsi akan menggunakan sintaks `PASSWORD_BCRYPT` yang menggunakan tipe
 15 *hash* berupa *CRYPT_BLOWFISH*. Selain itu, terdapat fungsi untuk melakukan pengecekan *password*
 16 yang sudah dienkripsi. Berikut merupakan contoh pengubahan kode untuk melakukan pengecekan
 17 *password* yang sudah dienkripsi.

18 `password_verify($password, $query->getRow()->password)`

19 Sintaks diatas menerima dua buah parameter dengan parameter pertama berupa masukan dari
 20 pengguna dan parameter berikutnya merupakan *hash* dari *password* yang sudah disimpan. Fungsi ini
 21 akan mengembalikan data berupa *true* apabila *password* sama dan *false* apabila *password* berbeda.

22 *Library* yang terdapat pada *CodeIgniter 4* juga dapat *diextend* dan dibentuk sesuai dengan
 23 kebutuhan. Berikut merupakan *library* yang dibentuk oleh pengguna dan akan dipindahkan menuju
 24 `app/Libraries`.

25 ***Twig***

26 *Library* ini tidak akan digunakan untuk membentuk *view* pada *CodeIgniter 4* namun, akan ada
 27 penggunaan sebuah fungsi *Twig* yang akan dibentuk pada direktori `app/Libraries`. Fungsi tersebut
 28 bernama `extra_time_formatter` yang memiliki fungsi untuk mengubah input yang diberikan
 29 menjadi format jam dikali enam puluh menit. Kode 4.34 merupakan fungsi yang akan dibentuk
 30 pada direktori `app/Libraries` dengan nama `Twig.php`.

Kode 4.34: Perancangan perubahan *library MY_Form_validation* pada *CodeIgniter 4*

```
32 1 <?php
33 2 namespace App\Libraries;
34 3
35 4 class Twig
36 5 {
37 6
38 7     /**
39 8      * Required
40 9      *
41 10     * @param  string
42 11     * @return bool
43 12 }
```

```

12     */
13     public function extra_time_formatter($extra_time)
14     {
15         // convert to minutes
16         $extra_time = floor($extra_time/60);
17         // convert to H*60
18         if ($extra_time % 60 == 0 )
19             $extra_time = ($extra_time/60) . '*60';
20         return $extra_time;
21     }
22 }
```

13 Kode 4.34 merupakan fungsi yang akan dibentuk pada file `Twig.php` dan akan digunakan pada
14 halaman `add_assignment.php`.

15 *MY_Form_validation*

16 *Library MY_Form_validation* akan dipindahkan menuju direktori `app/Libraries`. *Library* ini
17 akan digunakan kembali dengan perubahan `extends` menjadi menuju `Validation`, penghapusan
18 sintaks `defined`, dan akan ada penambahan `namespace` pada baris awal file. Kode 3.22 merupakan
19 contoh penambahan `namespace` dan penggantian `extends` pada *library* ini.

Kode 4.35: Contoh perubahan *library MY_Form_validation* pada *CodeIgniter 4*

```

20 namespace App\Libraries;
21 1
22 2
23 3 use CodeIgniter\Validation\Validation;
24 4
25 5 class MY_Form_validation extends Validation
```

27 Kode 4.35 menghapus sintaks `defined` dan menggantikannya dengan penambahan `namespace`. Selain
28 itu, kelas *library* akan `extends Validation`.

29 *Parsedown*

30 *Library Parsedown* akan dipindahkan menuju direktori `app/Libraries`. *Library* ini akan digunakan
31 kembali dengan penambahan `namespace` pada baris awal file dan penghapusan sintaks `defined`.
32 Kode 4.36 merupakan contoh penambahan `namespace` dan juga penambahan sintaks `defined`.

Kode 4.36: Perancangan perubahan *library Parsedown* pada *CodeIgniter 4*

```

33 1 namespace App\Libraries;
34 2
35 3 class Parsedown
```

38 Kode 4.36 menghapus sintaks `defined` dan menggantikannya dengan penambahan `namespace`.
39 Penggunaan *library* ini tidak akan berubah sehingga tidak terdapat perbedaan sintaks. Namun,
40 terdapat perubahan cara inisiasi *library* ini dimana sekarang akan menggunakan sintaks `new` dan
41 dilakukan inisiasi pada `BaseController`. Kode 4.37 merupakan perubahan cara melakukan inisiasi
42 *library parsedown*.

Kode 4.37: Perancangan perubahan inisiasi *library Parsedown* pada *controller* *CodeIgniter 4*

```

43 1 protected $parsedown;
44 2
45 3 /**
46 4 * Constructor.
47 5 */
48 6 public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
49 7 {
50 8     // Do Not Edit This Line
51 9     parent::initController($request, $response, $logger);
```

```

10    // Preload any models, libraries, etc, here.
11
12    // E.g.: $this->session = \Config\Services::session();
13    $this->parsedown = new Parsedown();
14
15 }

```

8 Kode 4.37 merupakan perubahan inisiasi pada *CodeIgniter 4*. *Library parsedown* akan dilakukan
 9 inisiasi menuju variabel **parsedown** yang sudah dibentuk pada luar fungsi. Inisiasi dilakukan pada
 10 *BaseController* karena terdapat pemakaian pada beberapa *model* dan *controller*.

11 ***Phpxcel***

12 *Library* ini akan digunakan kembali namun tidak akan dipindahkan menuju *app/Libraries*. *Library*
 13 akan dilakukan instalasi melalui *composer* dengan sintaks berikut:

```
14      composer require phpoffice/phpexcel
```

15 Sintaks diatas akan dijalankan pada akar dari aplikasi dan tidak terdapat perubahan terhadap
 16 penggunaan sintaks ini.

17 ***Shj_pagination***

18 *Library* ini akan digunakan kembali dan dipindahkan menuju direktori *app/Libraries*. Selain itu,
 19 terdapat penambahan *namespace* pada baris awal file dan penghapusan sintaks *defined*. Kode 4.38
 20 merupakan penambahan perubahan dan penambahan sintaks pada *library* ini.

Kode 4.38: Perancangan perubahan *library Shj_pagination* pada *CodeIgniter 4*

```

21 namespace App\Libraries;
22
23 class Shj_pagination
24 {
25
26 }

```

27 Kode 4.38 merupakan perubahan sintaks dalam membentuk *library* ini. Selanjutnya *library*
 28 dapat dipanggil menggunakan sintaks berikut.

```
29      $shj_pagination = new Shj_pagination($config);
```

30 Sintaks diatas akan memanggil *library* dan memasukan konfigurasi yang telah dibentuk.

31 **4.2.6 *Models***

32 Direktori ini akan berisikan seluruh *model* yang dipindahkan dari *CodeIgniter 3*. Berikut merupakan
 33 isi pada direktori ini:

- 34 • AssignmentModel.php
- 35 • HofModel.php
- 36 • LogsModel.php
- 37 • NotificationsModel.php
- 38 • QueueModel.php
- 39 • ScoreboardModel.php
- 40 • SettingsModel.php
- 41 • SubmitModel.php

- 1 • `User.php`
- 2 • `UserModel.php`

3 Seluruh *Model* akan diganti penamaannya dari yang sebelumnya menggunakan *snakecase* menjadi
4 *PascalCase*. Kode 4.39 merupakan perubahan sintaks yang terdapat pada *model*.

Kode 4.39: Perancangan perubahan *model* pada *CodeIgniter 4*

```

5  namespace App\Models;
6 1
7 2
8 3 use App\Libraries\Parsedown;
9 4 use CodeIgniter\Model;
10 5 use App\Models\SettingsModel;
11 6 use App\Models\ScoreboardModel;
12 7
13 8 class AssignmentModel extends Model
14 9 {
15 10     protected $settings_model;
16 11     protected $scoreboard_model;
17 12     protected $request;
18 13     protected $parsedown;
19 14
20 15     public function __construct()
21 16     {
22 17         parent::__construct();
23 18         $this->settings_model = new SettingsModel();
24 19         $this->request = \Config\Services::request();
25 20     }
26 21
27 22 /**
28 23 * Add New Assignment to DB / Edit Existing Assignment
29 24 */
30 25 * @param $id
31 26 * @param bool $edit
32 27 * @return bool
33 28 */
34 29 public function add_assignment($id, $edit = FALSE)
35 30 {
36 31     $this->scoreboard_model = new ScoreboardModel();
37 32 }
```

38 Kode 4.39 merupakan perubahan *model* yang terdapat pada *CodeIgniter 4*. Terdapat penghapusan sintaks `defined` dan juga terdapat perubahan dalam `extends` dimana sekarang akan `extends Model`. Selain itu terdapat pemanggilan *model* lainnya menggunakan sintaks `new` yang disimpan melalui variabel yang sudah dibentuk. Terdapat juga perubahan cara pengambilan data menggunakan fungsi *database* yang dapat dilihat pada kode 4.40.

Kode 4.40: Perubahan sintaks pada *model*

```

43
44 1 public function all_assignments()
45 2 {
46 3     $result = $this->db->table('assignments')->orderBy('id')->get()->getResultArray();
47 4     $assignments = [];
48 5     foreach ($result as $item)
49 6     {
50 7         $assignments[$item['id']] = $item;
51 8     }
52 9     return $assignments;
53 10 }
```

55 Kode 4.40 merupakan perubahan sintaks untuk melakukan pengambilan menggunakan fungsi *database*. Pengambilan data pada *CodeIgniter 4* perlu mendefinisikan nama tabel yang akan diambil datanya. Selanjutnya sintaks akan bersifat sama namun terdapat perubahan dari yang sebelumnya *snakecase* menjadi *camelCase*. *Model* yang telah dibentuk dapat dipanggil menuju *controller* menggunakan sintaks berikut.

60 \$this->settings_model = new SettingsModel();

61 Sintaks diatas akan memanggil *SettingsModel* dan disimpan menuju variabel. Fungsi *database*
62 juga digunakan file lain dengan melakukan inisiasi dengan sintaks berikut.

```
1                     $this->db = db_connect();  
2 Sintaks diatas akan melakukan inisiasi fungsi database dan menyimpannya menuju variabel yang  
3 telah dibentuk.
```

4 4.2.7 View

```
5 Direktori ini berisikan seluruh view yang dipindahkan dari CodeIgniter 3. Berikut merupakan isi  
6 dari direktori ini:
```

7 errors

```
8 Firektori ini berisikan file default dari CodeIgniter 4 karena tidak terdapat perubahan pada SharIF  
9 Judge CodeIgniter 3.
```

10 pages

```
11     • assignments.php  
12     • dashboard.php  
13     • halloffame.php  
14     • notifications.php  
15     • problems.php  
16     • profile.php  
17     • scoreboard_table.php  
18     • scoreboard.php  
19     • submissions.php  
20     • submit.php
```

```
21 Direktori admin:
```

```
22     • add_assignment.php  
23     • add_notification.php  
24     • add_user_result.php  
25     • add_user.php  
26     • delete_assignment.php  
27     • edit_problem_html.php  
28     • edit_problem_md.php  
29     • edit_problem_plain.php  
30     • install.php  
31     • logs.php  
32     • moss.php  
33     • queue.php  
34     • rejudge.php  
35     • settings.php  
36     • users.php
```

```
37 Direktori authentication:
```

```
38     • login.php
```

- lost.php
 - register_success.php
 - register.php
 - reset_password.php

5 Direktori templates:

- base.php
 - side_bar.php
 - simple_header.php
 - top_bar.php

10 View akan diubah menggunakan *extension .php* karena *twig* sudah tidak terintegrasi *CodeIgniter*

11 4. Seluruh file view akan diubah menjadi extension .php dari yang sebelumnya menggunakan .twig.

¹² Seluruh *delimiters* juga akan diubah menggunakan fungsi pada *CodeIgniter 4*. Perubahan *view* dapat dilihat pada kode [4.41](#).

Kode 4.41: Perubahan *view* pada *Login.php*

```
14<!-- {#
15 # SharIF Judge
16 # file: login.twig
17 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
18 #} -->
19 <!DOCTYPE html>
20 <html lang="en">
21 <?= $this->include('templates/simple_header')?>
22
23 <?= form_open() ?>
24     <div class="box login">
25
26         <div class="judge_logo">
27             <a href="<?= site_url() ?>"></a>
28         </div>
29
30
31         <div class="login_form">
32             <div class="login1">
33                 <p>
34                     <label for="form_username">Username</label><br/>
35                     <input id="form_username" type="text" name="username" required="required" pattern="[0-9a-zA-Z]{3,20}" title="The
36                         Username field must be between 3 and 20 characters in length, and contain only digits and lowercase
37                         letters" class="sharif_input" value="<?= set_value('username') ?>" autofocus="autofocus"/>
38                     <?= isset($this->errors['username'])?>
39                 </p>
40
41                 <p>
42                     <label for="form_password">Password</label><br/>
43                     <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The
44                         Password field must be at least 6 characters in length" class="sharif_input"/>
45                     <?= isset($this->errors['password'])?>
46                 </p>
47
48                 <?php if ($error): ?>
49                     <div class="shj_error">Incorrect username or password.</div>
50                 <?php endif ?>
51
52             </div>
53             <div class="login2">
54                 <p style="margin:0;">
55                     <?php if ($registration_enabled): ?>
56                     <a href="<?= site_url('register') ?>">Register</a> |
57                     <?php endif ?>
58                     <a href="<?= site_url('login/lost') ?>">Reset Password</a>
59                     <input type="submit" value="Login" id="sharif_submit"/>
60
61                 </p>
62             </div>
63
64         </div>q
65 <?= form_close() ?>
66
67 </body>
68 </html>
```

Kode 4.41 merupakan perubahan yang terdapat pada halaman *view*. Delimiters {{ }} akan digantikan menjadi <?= ?> sedangkan {%-%} akan digantikan menjadi <?php ?>. Delimiters

yang memanggil fungsi pada *CodeIgniter 4* akan digantikan menjadi <?= ?>. Perubahan juga terdapat pada sintaks dari yang sebelumnya menggunakan `include` akan digantikan menggunakan fungsi *CodeIgniter 4* berupa `$this->include`. Selain terdapat perubahan *extension* dan *delimiters*, terdapat juga penambahan kode pada *Controller* karena tidak mendukung pembentukan variabel *global* pada *View*. Kode 4.42 merupakan contoh penambahan kode pada *Controller*.

Kode 4.42: Penambahan kode pada *Login.php*

```

6  $data = [
7  1     'error' => FALSE,
8  2     'registration_enabled' => $this->settings_model->get_setting('enable_registration'),
9  3     'title' => 'Login',
10 4     'validationError' => $this->validation
11 5   ];
12 6

```

Kode 4.42 merupakan contoh penambahan data pada *controller*. Penambahan data terjadi karena halaman *view* PHP tidak dapat mendeklarasikan variabel secara *global* sehingga data-data seperti *title* tidak dapat diakses oleh *view* lainnya.

4.2.8 public

assets

Direktori ini berisikan seluruh data yang dapat dilihat oleh pengguna seperti *javascript*, gambar, dan lainnya. Berikut merupakan isi pada direktori ini:

- ace
- font
- fullcalendar
- gridster
- images
- js
- nano_scroller
- noty
- pdfjs
- reveal
- snippet
- styles
- tinymce

Direktori *assets* akan dipindah tanpa ada perubahan apapun.

4.2.9 restricted

Direktori ini akan dipindah tanpa ada perubahan apapun.

1

BAB 5

2

IMPLEMENTASI DAN PENGUJIAN

3 Bab ini membahas mengenai implementasi dan pengujian konversi *SharIF Judge*.

4 **5.1 Lingkungan Implementasi dan Pengujian**

5 Implementasi perangkat lunak *SharIF Judge* dilakukan pada dua buah lingkungan. Lingkungan
6 pertama digunakan untuk membangun perangkat lunak sedangkan lingkungan kedua digunakan
7 untuk melakukan pengujian. Berikut merupakan spesifikasi lingkungan implementasi dan pengujian
8 yang digunakan:

9 1. Lingkungan *Development*

10 Tabel 5.1 merupakan spesifikasi perangkat keras lingkungan *development*.

Tabel 5.1: Perangkat Keras Lingkungan *Development*

Parameter	Nilai
<i>Processor</i>	<i>M1 Pro</i>
<i>Random Access Memory (RAM)</i>	16 GB
<i>Storage</i>	512 GB <i>SSD</i>

11 Tabel 5.2 merupakan spesifikasi perangkat lunak lingkungan *development*.

Tabel 5.2: Perangkat Lunak Lingkungan *Development*

Parameter	Nilai
Sistem Operasi	Debian
Bahasa Pemrograman	<i>PHP, JavaScript, CSS, dan HTML</i>
Framework	<i>CodeIgniter 4.2.3</i>
Code Editor	<i>Visual Studio Code 1.84.2 (Universal)</i>
Perangkat Lunak Pendukung	<i>Docker</i> <i>Composer 2.6.5</i> <i>Google Chrome Version 119.0.6045.159 (Official Build) (arm64)</i> <i>MySQL</i>

12 2. Lingkungan *Staging*

13 Tabel 5.3 merupakan spesifikasi perangkat keras lingkungan *staging*.

Tabel 5.3: Perangkat Keras Lingkungan *staging*

Parameter	Nilai
<i>Processor</i>	<i>M1 Pro</i>
<i>Random Access Memory (RAM)</i>	512 MB
<i>Storage</i>	10 GB

¹ Tabel 5.4 merupakan spesifikasi perangkat lunak lingkungan *staging*.

Tabel 5.4: Perangkat Lunak Lingkungan *Staging*

Parameter	Nilai
Sistem Operasi	Ubuntu 20.04
Bahasa Pemrograman	PHP, <i>JavaScript</i> , <i>CSS</i> , dan <i>HTML</i>
<i>Perangkat Lunak Pendukung</i>	<i>Apache Server</i> <i>Composer</i> <i>MySQL</i> PHP 8.1

² 5.2 Implementasi

³ Berikut merupakan hasil implementasi dari analisis dan perancangan perangkat lunak yang telah
⁴ dibentuk:

⁵ 5.2.1 Kode Program

⁶ Perubahan dan penambahan kode program untuk melakukan konversi perangkat lunak telah
⁷ dijabarkan pada bab 4. Seluruh perubahan dan penambahan dibentuk menggunakan bahasa
⁸ pemrograman PHP dengan bantuan beberapa perangkat lunak tambahan. Kode program untuk
⁹ *controller* dapat dilihat pada A. Kode program untuk *model* dapat dilihat pada B. Kode program
¹⁰ untuk *view* dapat dilihat pada C.

¹¹ 5.2.2 Basis Data

¹² Basis data terdapat perubahan pada tabel *session* dimana terdapat perubahan nama kolom dan
¹³ penghapusan kolom. Terdapat perubahan nama kolom *session_id* menjadi *id*, *user_data* menjadi
¹⁴ *data*, dan *last_activity* menjadi *timestamp*. Terdapat penghapusan kolom *user_agent*, perubahan
¹⁵ tipe data kolom *timestamp* dan *data*. Tabel 5.5 merupakan struktur tabel baru pada *shj_sessions*.

Tabel 5.5: Struktur baru tabel *shj_session*

Atribut	Tipe Data	Ukuran	Default
<i>id</i> (<i>primary key</i>)	varchar	128	None
<i>ip_address</i>	varchar	45	None
<i>timestamp</i>	timestamp	-	current_timestamp()
<i>data</i>	blob	-	None

5.3 Pengujian Fungsional

- 2 Pengujian fungsional dilakukan pada lingkungan kedua yang bertujuan untuk melakukan pengujian
3 terhadap seluruh fungsi pada perangkat lunak *SharIF Judge*. Pengujian dilakukan untuk memastikan
4 setiap fitur yang terdapat pada *SharIF Judge* versi *CodeIgniter 4* dapat berjalan dengan baik setelah
5 dilakukan konversi. Berikut merupakan pengujian terhadap fitur-fitur pada *SharIF Judge*:
- 6 1. Melakukan instalasi *SharIF Judge*
7 • Reaksi yang diharapkan: *Database* terbentuk dan lkun *admin* tersimpan pada *database*.
8 • Reaksi: *Database* terbentuk dan lkun *admin* tersimpan pada *database*.
- 9 2. Melakukan *login*
10 • Reaksi yang diharapkan: Pengguna masuk menuju *dashboard* menggunakan akun yang
11 telah dibentuk.
12 • Reaksi: Pengguna masuk menuju *dashboard* dengan akun yang telah dibentuk.
- 13 3. Melakukan *login* dengan *password* yang salah
14 • Reaksi yang diharapkan: Tidak dapat masuk dan mengembalikan *error message* berupa
15 *Incorrect username or password*.
16 • Reaksi: Pengguna tidak dapat masuk dan aplikasi mengembalikan *error message* berupa
17 *Incorrect username or password*.
- 18 4. Melakukan *register*
19 • Reaksi yang diharapkan: Pengguna berhasil melakukan *register* dan aplikasi mengembalikan
20 *Registered successfully*.
21 • Reaksi: Pengguna berhasil melakukan *register* dan aplikasi mengembalikan *Registered*
22 *successfully*.
- 23 5. Melakukan *reset password* dengan *email* yang telah terdaftar
24 • Reaksi yang diharapkan: Dikirimkan *email* berupa *link* untuk melakukan *reset password*,
25 pengguna diarahkan menuju situs *SharIF Judge*, dan pengguna memasukan *password*
26 baru untuk *login*.
27 • Reaksi: Dikirimkan *email* berupa *link* untuk melakukan *reset password*, pengguna dia-
28 rahkan menuju situs *SharIF Judge*, dan pengguna dapat memasukan *password* baru
29 untuk *login*.
- 30 6. *Top bar* pengguna *admin*
31 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown tools*, *dropdown select assignments*,
32 *timer assignment*, dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*,
33 dan *cheat detection*.
34 • Reaksi: *Top bar* memiliki *dropdown tools*, *dropdown select assignments*, *timer assignment*,
35 dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*, dan *cheat detection*.
- 36 7. *Top bar* pengguna *head instructor*
37 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown tools*, *dropdown select assignments*,
38 *timer assignment*, dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*,
39 dan *cheat detection*.
40 • Reaksi: *Top bar* terdapat *dropdown tools*, *dropdown select assignments*, *timer assignment*,
41 dan *icon profile*. *Dropdown tools* berisikan *rejudge*, *submission queue*, dan *cheat detection*.
- 42 8. *Top bar* pengguna *instructor*

- 1 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown select assignments, timer assignment,*
2 dan *icon profile.*
3 • Reaksi: *Top bar* terdapat *dropdown select assignments, timer assignment,* dan *icon*
4 *profile.*

5 9. *Top bar pengguna student*

- 6 • Reaksi yang diharapkan: *Top bar* terdapat *dropdown select assignments, timer assignment,*
7 dan *icon profile.*
8 • Reaksi: *Top bar* terdapat *dropdown select assignments, timer assignment,* dan *icon*
9 *profile.*

10 10. *Side bar pengguna admin*

- 11 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard, settings, users, notifications,*
12 *assignments, problems, submit, final submissions, all submissions, scoreboard, hall of*
13 *fame, dan 24-hour log.*
14 • Reaksi: *Sidebar* berisikan *dashboard, settings, users, notifications, assignments, problems,*
15 *submit, final submissions, all submissions, scoreboard, hall of fame, dan 24-hour log.*

16 11. *Side bar pengguna head instructor*

- 17 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard, notifications, assignments, pro-*
18 *blems, submit, final submissions, all submissions, scoreboard, dan hall of fame.*
19 • Reaksi: *Sidebar* berisikan *dashboard, notifications, assignments, problems, submit, final*
20 *submissions, all submissions, scoreboard, dan hall of fame.*

21 12. *Side bar pengguna instructor*

- 22 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard, notifications, assignments, pro-*
23 *blems, submit, final submissions, all submissions, scoreboard, dan hall of fame.*
24 • Reaksi: *Sidebar* berisikan *dashboard, notifications, assignments, problems, submit, final*
25 *submissions, all submissions, scoreboard, dan hall of fame.*

26 13. *Side bar pengguna student*

- 27 • Reaksi yang diharapkan: *Sidebar* berisikan *dashboard, notifications, assignments, pro-*
28 *blems, submit, final submissions, all submissions, scoreboard, dan hall of fame.*
29 • Reaksi: *Sidebar* berisikan *dashboard, notifications, assignments, problems, submit, final*
30 *submissions, all submissions, scoreboard, dan hall of fame.*

31 14. *Dashboard pengguna admin dan head instructor*

- 32 • Reaksi yang diharapkan: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments*
33 dan *latest notification* dengan *icon* untuk menambahkan *notification.*
34 • Reaksi: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments* dan *latest*
35 *notification* dengan *icon* untuk menambahkan *notification.*

36 15. *Dashboard pengguna instructor dan student*

- 37 • Reaksi yang diharapkan: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments*
38 dan *latest notification.*
39 • Reaksi: *Dashboard* berisikan *calendar* dengan jangka waktu *assignments* dan *latest*
40 *notification.*

41 16. *Profile pengguna admin*

- 42 • Reaksi yang diharapkan: Berisikan *username, name, email, password, password again,*

1 dan *user role*.

- 2 • Reaksi: Berisikan *username*, *name*, *email*, *password*, *password again*, dan *user role*.

3 17. *Profile* pengguna *head instructor*, *instructor*, dan *student*

- 4 • Reaksi yang diharapkan: Berisikan *username*, *name*, *email*, *password*, dan password
5 again.

- 6 • Reaksi: Berisikan *username*, *name*, *email*, *password*, dan password again.

7 18. Mengubah *password* akun pada *profile*

- 8 • Reaksi yang diharapkan: *Password* pada akun berubah dan dapat dilakukan oleh seluruh
9 pengguna.

- 10 • Reaksi: *Password* pada akun berubah dan dapat dilakukan oleh seluruh pengguna.

11 19. Mengubah *role*

- 12 • Reaksi yang diharapkan: Penggantian *role* hanya dapat dilakukan oleh *admin*.

- 13 • Reaksi: Penggantian *role* hanya dapat dilakukan oleh *admin* dan akan mengembalikan
14 *error* apabila pengguna bukan *admin*.

15 20. Halaman *settings*

- 16 • Reaksi yang diharapkan: Halaman hanya dapat diakses oleh *admin* dan akan mengelu-
17 arkan *error message* apabila menggunakan *role* lain.

- 18 • Reaksi: Halaman hanya dapat diakses oleh *admin* dan mengeluarkan *error message*
19 berupa 404 apabila diakses oleh *role* lain.

20 21. Melakukan *sign out*

- 21 • Reaksi yang diharapkan: Pengguna kembali ke halaman *login* dan *session* pengguna
22 dihancurkan.

- 23 • Reaksi: Pengguna kembali ke halaman *login* dan *session* pengguna dihancurkan.

24 22. Mengubah konfigurasi pada halaman *settings*

- 25 • Reaksi yang diharapkan: Konfigurasi yang diubah disimpan menuju *database*.

- 26 • Reaksi: Konfigurasi yang diubah disimpan menuju *database*

27 23. Menyalakan fitur *registration*

- 28 • Reaksi yang diharapkan: Halaman *login* terdapat tombol *register* dan pengguna dapat
29 melakukan *register*.

- 30 • Reaksi: Halaman *login* terdapat tombol *register* dan pengguna dapat melakukan *register*.

31 24. Menyalakan fitur *lock student display name*

- 32 • Reaksi yang diharapkan: Fitur untuk mengubah *display name* telah *disable*.

- 33 • Reaksi: Fitur untuk mengubah *display name* telah *disable*.

34 25. Halaman *users*

- 35 • Reaksi yang diharapkan: Halaman hanya dapat diakses oleh *admin*.

- 36 • Reaksi: Halaman hanya dapat diakses oleh *admin* dan akan mengembalikan *error* berupa
37 404 apabila pengguna bukan *admin*.

38 26. Menambah *users* dan mengirimkan email kepada setiap pengguna

- 39 • Reaksi yang diharapkan: Pengguna dapat ditambahkan lebih dari satu dan email dikiri-
40 rimkan berisikan data pengguna.

- 41 • Reaksi: Pengguna bertambah dan email dikirimkan berisikan data dari pengguna berupa
42 *username* dan *password*.

1 27. Mengunduh pengguna dalam format *excel*

- 2 • Reaksi yang diharapkan: Seluruh data pada halaman *users* terunduh pada perangkat
3 pengguna dalam format *excel*.
4 • Reaksi: Seluruh data pada halaman *users* terunduh pada perangkat pengguna dalam
5 format *excel*.

6 28. Mengganti data pengguna lain

- 7 • Reaksi yang diharapkan: Data pengguna lain hanya dapat diubah oleh *admin*. Data
8 berubah dan pengguna dapat mengakses menggunakan data yang baru.
9 • Reaksi: Data pengguna lain hanya dapat diubah oleh *admin* dan akan mengembalikan
10 *error* berupa 404 apabila pengguna bukan *admin*. Data berubah dan pengguna dapat
11 mengakses menggunakan data yang baru.

12 29. Melihat seluruh pengumpulan setiap pengguna

- 13 • Reaksi yang diharapkan: Pengguna *student* hanya dapat melihat seluruh pengumpulannya.
14 Pengguna *admin*, *head instructor*, dan *instructor* dapat melihat pengumpulan pengguna
15 lain.
16 • Reaksi: Pengguna *student* hanya dapat melihat seluruh pengumpulannya. Pengguna
17 *admin*, *head instructor*, dan *instructor* dapat melihat pengumpulan pengguna lain.

18 30. Menghapus pengguna

- 19 • Reaksi yang diharapkan: Hanya dapat dilakukan oleh *admin* dan pengguna terhapus
20 dari *database*.
21 • Reaksi: Hanya dapat dilakukan oleh *admin* dan apabila dilakukan oleh pengguna lain
22 akan mengembalikan *error* 404. Data pengguna terhapus dari *database*.

23 31. Menghapus *submission* pengguna

- 24 • Reaksi yang diharapkan: Seluruh *submission* dari pengguna tersebut terhapus.
25 • Reaksi: Seluruh *submission* dari pengguna tersebut terhapus.

26 32. Halaman *notifications*

- 27 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna namun tombol *add*, *edit*
28 dan *delete* hanya terdapat pada pengguna *admin* dan *head instructor*.
29 • Reaksi: Dapat diakses oleh seluruh pengguna namun tombol *add*, *edit* dan *delete* hanya
30 terdapat pada pengguna *admin* dan *head instructor*.

31 33. Halaman *assignments*

- 32 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna dan dapat memilih
33 *assignment*.
34 • Reaksi: Dapat diakses oleh seluruh pengguna dan dapat memilih *assignment*.

35 34. Memilih *assignment*

- 36 • Reaksi yang diharapkan: Waktu pada *top bar* berganti menjadi penghitung mundur
37 waktu *assignment* selesai. Apabila terdapat *extra time* maka muncul tampilan dan
38 penghitung mundur *extra time*.
39 • Reaksi: Waktu pada *top bar* berganti menjadi penghitung mundur waktu *assignment*
40 selesai. Apabila terdapat *extra time* maka muncul tampilan dan penghitung mundur
41 *extra time*.

42 35. Menambah *assignment*

- 1 • Reaksi yang diharapkan: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Assignment* bertambah pada direktori *assignments* dan data ter*extract*. Dapat menambahkan *problem* dan melakukan konfigurasi sesuai dengan kebutuhan.
- 2 • Reaksi: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Assignment* bertambah pada direktori *assignments* dan data ter*extract*. Dapat menambahkan *problem* dan melakukan konfigurasi sesuai dengan kebutuhan.

3 36. Mengunduh *pdf* soal

- 4 • Reaksi yang diharapkan: *PDF* soal terunduh menuju perangkat pengguna.
- 5 • Reaksi: *PDF* soal terunduh menuju perangkat pengguna.

6 37. Mengunduh *test case* dan soal *PDF*

- 7 • Reaksi yang diharapkan: *Test case* dan soal *pdf* terunduh menuju perangkat pengguna dengan *extension zip*.
- 8 • Reaksi: *Test case* dan soal *pdf* terunduh menuju perangkat pengguna dengan *extension zip*.

9 38. Mengunduh *final submission* berdasarkan pengguna

- 10 • Reaksi yang diharapkan: Seluruh *final submission* dari setiap pengguna yang mengumpulkan terunduh pada perangkat dengan *extension zip*. Seluruh file yang terunduh dikelompokan pada direktori berdasarkan pengguna.
- 11 • Reaksi: Seluruh *final submission* dari setiap pengguna yang mengumpulkan terunduh pada perangkat dengan *extension zip*. Seluruh file yang terunduh dikelompokan pada direktori berdasarkan pengguna.

12 39. Mengunduh *final submission* berdasarkan *problem*

- 13 • Reaksi yang diharapkan: Seluruh *final submission* dari pengguna yang mengumpulkan terunduh pada perangkat dengan *extension zip*. Seluruh file yang terunduh dikelompokan pada direktori berdasarkan *problem*.
- 14 • Reaksi: Seluruh *final submission* dari pengguna yang mengumpulkan terunduh pada perangkat dengan *extension zip*. Seluruh file yang terunduh dikelompokan pada direktori berdasarkan *problem*.

15 40. Mendeteksi *similar code* pada *assignment*

- 16 • Reaksi yang diharapkan: Pengguna diarahkan menuju halaman *cheat detection* dengan *id* dari *assignment* yang dipilih. Dihasilkan tautan yang berisikan hasil dari pengecekan *similar code*.
- 17 • Reaksi: Pengguna diarahkan menuju halaman *cheat detection* dengan *id* dari *assignment* yang dipilih. Dihasilkan tautan yang berisikan hasil dari pengecekan *similar code*.

18 41. Melakukan *edit assignment*

- 19 • Reaksi yang diharapkan: *Assignment* berubah sesuai dengan konfigurasi pengguna.
- 20 • Reaksi: *Assignment* berubah sesuai dengan konfigurasi pengguna.

21 42. Menghapus *assignment*

- 22 • Reaksi yang diharapkan: *Assignment* ter*hapus* dari halaman *assignments*.
- 23 • Reaksi: *Assignment* ter*hapus* dari halaman *assignments*.

24 43. Halaman *problems*

- 25 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna dan pengguna dapat

1 mengumpukan jawaban melalui halaman tersebut. Namun, dekripsi *problem* hanya dapat
2 dilakukan oleh *admin* dan *head instructor*.

- 3 • Reaksi: Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna dan pengguna
4 dapat mengumpukan jawaban pada halaman tersebut. Namun, dekripsi *problem* hanya
5 dapat dilakukan oleh *admin* dan *head instructor*.

6 44. Menambahkan deskripsi *problem* menggunakan *markdown*

- 7 • Reaksi yang diharapkan: Deskripsi *problem* ditambahkan dengan tipe bahasa *markdown*.
8 Deskripsi ditampilkan pada halaman *problem*.
9 • Reaksi: Deskripsi *problem* ditambahkan dengan tipe bahasa *markdown*. Deskripsi
10 ditampilkan pada halaman *problem*.

11 45. Menambahkan deskripsi *problem* menggunakan *HTML*

- 12 • Reaksi yang diharapkan: Deskripsi *problem* ditambahkan dengan tipe bahasa *HTML*.
13 Deskripsi ditampilkan pada halaman *problem*.
14 • Reaksi: Deskripsi *problem* ditambahkan dengan tipe bahasa *HTML*. Deskripsi ditampilkan
15 pada halaman *problem*.

16 46. Menambahkan deskripsi *problem* menggunakan *plain HTML*

- 17 • Reaksi yang diharapkan: Deskripsi *problem* ditambahkan dengan tipe bahasa *plain*
18 *HTML*. Deskripsi ditampilkan pada halaman *problem*.
19 • Reaksi: Deskripsi *problem* ditambahkan dengan tipe bahasa *plain HTML*. Deskripsi ditampilkan
20 pada halaman *problem*.

21 47. Halaman *submit*

- 22 • Reaksi yang diharapkan: Dapat diakses oleh seluruh pengguna. Pengguna dapat memilih
23 *problem* dan bahasa pemrogramannya. *PDF* soal ditampilkan. Pengguna dapat langsung
24 mencoba dan mengumpulkan kode melalui *editor* kode.
25 • Reaksi: Dapat diakses oleh seluruh pengguna. Pengguna dapat memilih *problem* dan
26 bahasa pemrogramannya. *PDF* soal ditampilkan. Pengguna dapat langsung mencoba
27 dan mengumpulkan kode melalui *editor* kode.

28 48. Halaman *final submissions*

- 29 • Reaksi yang diharapkan: Melihat seluruh *final submission* dari seluruh pengguna apabila
30 pengguna berupa *admin*, *head instructor*, dan *instructor*. Melihat seluruh *final submission*
31 dari pengguna apabila pengguna *student*.
32 • Reaksi: Melihat seluruh *final submission* dari seluruh pengguna apabila pengguna berupa
33 *admin*, *head instructor*, dan *instructor*. Melihat seluruh *final submission* dari pengguna
34 apabila pengguna *student*.

35 49. Mengunduh data *final submission* dalam format *excel*

- 36 • Reaksi yang diharapkan: Seluruh data dari halaman *final submission* terunduh kepada
37 perangkat pengguna dalam format *excel*.
38 • Reaksi: Seluruh data dari halaman *final submission* terunduh kepada perangkat pengguna
39 dalam format *excel*.

40 50. Melihat status dari *submission* dan *test case problem*

- 41 • Reaksi yang diharapkan: Memperlihatkan status dan nilai dari kode. Setiap *test case*
42 memperlihatkan apakah benar, salah, atau *runtime error*

- 1 • Reaksi: Memperlihatkan status dan nilai dari kode. Setiap *test case* memperlihatkan
2 apakah benar, salah, atau *runtime error*

3 51. Melakukan *filter* berdasarkan nama pengguna

- 4 • Reaksi yang diharapkan: Hanya terlihat *final submissions* dari pengguna tersebut.
5 • Reaksi: Hanya terlihat *final submissions* dari pengguna tersebut.

6 52. Melakukan *filter* berdasarkan *problem*

- 7 • Reaksi yang diharapkan: Hanya terlihat *final submissions* dari *problem* tersebut.
8 • Reaksi: Hanya terlihat *final submissions* dari *problem* tersebut.

9 53. Melakukan *rejudge*

- 10 • Reaksi yang diharapkan: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Submission*
11 akan dilakukan penilaian ulang dan mengubah status sementara menjadi *pending*.
12 • Reaksi: Hanya dapat dilakukan oleh *admin* dan *head instructor*. *Submission* akan
13 dilakukan penilaian ulang dan mengubah status sementara menjadi *pending*.

14 54. Halaman *all submissions*

- 15 • Reaksi yang diharapkan: Melihat seluruh *submission* yang telah dikumpulkan dan dapat
16 memilih *submission* yang ingin dijadikan hasil akhir. Melihat status dari *submission*
17 pengguna. Melihat kode yang sudah dikumpulkan. Melihat *log* dari setiap *submission*
18 apabila pengguna berupa *admin*, *head instructor*, dan *instructor*.
19 • Reaksi: Melihat seluruh *submission* yang telah dikumpulkan dan dapat memilih *submission*
20 yang ingin dijadikan hasil akhir. Melihat status dari *submission* pengguna. Melihat
21 kode yang sudah dikumpulkan. Melihat *log* dari setiap *submission* apabila pengguna
22 berupa *admin*, *head instructor*, dan *instructor*.

23 55. Melakukan *filter* berdasarkan nama pengguna

- 24 • Reaksi yang diharapkan: Hanya terlihat *all submissions* dari pengguna tersebut.
25 • Reaksi: Hanya terlihat *all submissions* dari pengguna tersebut.

26 56. Melakukan *filter* berdasarkan *problem*

- 27 • Reaksi yang diharapkan: Hanya terlihat *all submissions* dari *problem* tersebut.
28 • Reaksi: Hanya terlihat *all submissions* dari *problem* tersebut.

29 57. Mengunduh data *all submission* dalam format *excel*

- 30 • Reaksi yang diharapkan: Seluruh data dari halaman *all submission* terunduh kepada
31 perangkat pengguna dalam format *excel*.
32 • Reaksi: Seluruh data dari halaman *all submission* terunduh kepada perangkat pengguna
33 dalam format *excel*.

34 58. Halaman *hall of fame*

- 35 • Reaksi yang diharapkan: Melihat skor total dari seluruh *assignment* setiap pengguna
36 dan menguratkannya dari yang terbesar.
37 • Reaksi: Melihat skor total dari seluruh *assignment* setiap pengguna dan menguratkannya
38 dari yang terbesar.

39 59. Halaman *24-hour log*

- 40 • Reaksi yang diharapkan: Hanya dapat diakses oleh *admin* dan berisikan seluruh data
41 perangkat dari pengguna. Pengguna yang melakukan *login* dari *ip address* yang berbeda
42 akan terdeteksi dan ditampilkan.

- 1 • Reaksi: Hanya dapat diakses oleh *admin* dan berisikan seluruh data perangkat dari
2 pengguna. Pengguna yang melakukan *login* dari *ip address* yang berbeda akan terdeteksi
3 dan ditampilkan

4 60. Halaman *submission queue*

- 5 • Reaksi yang diharapkan: Dapat memberhentikan, melanjutkan, dan membersihkan
6 antrian dari *submission* yang ada.
- 7 • Reaksi: Dapat memberhentikan, melanjutkan, dan membersihkan antrian dari *submission*
8 yang ada.

¹

BAB 6

²

KESIMPULAN DAN SARAN

³ Bab ini akan membahas kesimpulan yang telah diraih selama proses analisis, perancangan, im-
⁴ plementasi, pengujian yang telah dibentuk, serta saran-saran untuk pengembangan yang dapat
⁵ dipertimbangkan.

⁶ 6.1 Kesimpulan

⁷ Berdasarkan proses dan hasil analisis, perancangan, implementasi, dan pengujian perangkat lunak
⁸ yang telah dibentuk, maka diperoleh kesimpulan sebagai berikut:

- ⁹ 1. *SharIF Judge* telah berhasil dikonversi menuju *CodeIgniter 4*.
- ¹⁰ 2. Seluruh fitur yang terdapat pada *SharIF Judge CodeIgniter 4* berjalan sesuai dengan *SharIF*
¹¹ *Judge* pada *CodeIgniter 3*.
- ¹² 3. Terdapat perubahan penggunaan sintaks dan penghapusan beberapa *library* sehingga diganti
¹³ menggunakan fitur yang disediakan oleh PHP dan juga *library* yang terdapat pada *CodeIgniter*
¹⁴ *4*.
- ¹⁵ 4. Terdapat perubahan struktur *database* untuk menyesuaikan dengan *CodeIgniter 4*.

¹⁶ 6.2 Saran

¹⁷ Berdasarkan hasil analisis, perancangan, dan implementasi yang telah dilakukan, berikut merupakan
¹⁸ saran-saran untuk pengembangan selanjutnya:

- ¹⁹ 1. Membentuk *Sandbox* untuk melakukan *compile* bahasa pemrograman C dan CPP yang
²⁰ terintegrasi pada *Ubuntu* versi 22.
- ²¹ 2. Melakukan dokumentasi terhadap fitur-fitur yang telah dibentuk.
- ²² 3. Melakukan implementasi autentikasi *RADIUS* dan *LDAP*.

DAFTAR REFERENSI

- [1] Version 3.1.13 (2022) *CodeIgniter User Guide*. CodeIgniter Foundation. Richmond, Canada.
- [2] Version 1.4 (2023) *SharIF Judge Documentation*. Informatika UNPAR. Jl. Ciumbuleuit No. 94, Bandung.
- [3] Version 4.3 (2023) *CodeIgniter User Guide*. CodeIgniter Foundation. Richmond, Canada.
- [4] Prihatini, F. N. dan Indudewi, D. (2016) Kesadaran dan Perilaku Plagiarisme dikalangan Mahasiswa(Studi pada Mahasiswa Fakultas Ekonomi Jurusan Akuntansi Universitas Semarang). *Dinamika Sosial Budaya*, **18**, 68–75.
- [5] Kurnia, A., Lim, A., dan Cheang, B. (2001) Online judge. *Computers & Education*, **18**, 299–315.

LAMPIRAN A

KODE PROGRAM *CONTROLLERS*

Kode A.1: Assignments.php

```
1 <?php
2
3 namespace App\Controllers;
4
5 use App\Controllers\BaseController;
6 use App\Libraries\Unzip;
7 use App\Models\AssignmentModel;
8 use App\Models\SettingsModel;
9 use App\Models\SubmitModel;
10 use App\Models\User;
11 use ZipArchive;
12
13 /**
14 * SharIF Judge online judge
15 * @file Assignments.php
16 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
17 */
18
19 class Assignments extends BaseController
20 {
21
22     private $messages;
23     private $edit_assignment;
24     private $edit;
25     protected $assignment_model;
26     protected $validation;
27     protected $user;
28     protected $assignment;
29     protected $settings_model;
30     protected $zip;
31     protected $submit_model;
32     protected $unzip;
33     protected $parsedown;
34
35
36 // -----
37
38
39     public function __construct()
40     {
41         $this->messages = array();
42         $this->edit_assignment = array();
43         $this->edit = FALSE;
44         $this->assignment_model = new AssignmentModel();
45         $this->settings_model = new SettingsModel();
46         $this->validation = \Config\Services::validation();
47         $this->user = new User();
48     }
49
50
51 // -----
52
53
54     public function index()
55     {
56         $data = array(
57             'all_assignments' => $this->assignment_model->all_assignments(),
58             'messages' => $this->messages,
59             'selected' => 'assignments',
60             'user' => $this->user,
61             'finish_time' => $this->user->selected_assignment['finish_time'],
62             'extra_time' => $this->user->selected_assignment['extra_time'],
63         );
64
65         foreach ($data['all_assignments'] as &$item)
66         {
67             $extra_time = $item['extra_time'];
68             $delay = shj_now() - strtotime($item['finish_time']);
69             ob_start();
70             if ( eval($item['late_rule']) === FALSE )
71                 $coefficient = "error";
72             if (!isset($coefficient))
73                 $coefficient = "error";
74             ob_end_clean();
75             $item['coefficient'] = $coefficient;
76         }
77     }
78 }
```

```

76         $item['finished'] = ($delay > $extra_time);
77     }
78
79     return view('pages/assignments', $data);
80 }
81
82
83 // -----
84
85
86 /**
87 * Used by ajax request (for select assignment from top bar)
88 */
89
90 public function select()
91 {
92     if ( ! $this->request->isAJAX() )
93         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
94
95     $this->validation->setRule('assignment_select', 'Assignment', 'required|integer|greater_than[0]');
96
97     if ($this->validation->withRequest($this->request)->run())
98     {
99         $this->user->select_assignment($this->request->getPost('assignment_select'));
100        $this->assignment = $this->assignment_model->assignment_info($this->request->getPost('assignment_select'));
101        $json_result = array(
102            'done' => 1,
103            'finish_time' => $this->assignment['finish_time'],
104            'extra_time' => $this->assignment['extra_time'],
105        );
106    } else{
107        $json_result = array('done' => 0, 'message' => 'Input_Error');
108    }
109
110    $this->response->setHeader('Content-Type:', 'application/json; charset=utf-8');
111    echo json_encode($json_result);
112 }
113
114
115 // -----
116
117
118
119 /**
120 * Download pdf file of an assignment (or problem) to browser
121 */
122
123 public function pdf($assignment_id, $problem_id = NULL, $no_download = FALSE)
124 {
125     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
126     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
127     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
128
129     // Find pdf file
130     if ($problem_id === NULL || $problem_id === "null")
131         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/*.pdf";
132     else
133         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/p{$problem_id}/*.pdf";
134     $pdf_files = glob($pattern);
135     if ( ! $pdf_files )
136         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("File_not_found");
137     elseif ( ! $this->assignment_model->assignment_info($assignment_id)[ 'open' ] )
138         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("Selected_assignment_has_been_closed.");
139     elseif ( ! $this->assignment_model->is_participant($this->assignment_model->assignment_info($assignment_id)[ 'participants' ],
140             ',', $this->user->username) )
141         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("You_are_not_registered_for_submitting.");
142     elseif ( shj_now() > $finishtime + $extratime )
143         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("Selected_assignment_has_finished.");
144     elseif ( shj_now() < $starttime )
145         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound("Selected_assignment_has_not_started.");
146
147
148     $filename = shj_basename($pdf_files[0]);
149     // Download the file to browser
150     if($no_download === FALSE){
151         return $this->response->download($filename, file_get_contents($pdf_files[0]));
152     }
153     else{
154         $content = file_get_contents($pdf_files[0]);
155         header('Content-Type: application/pdf');
156         die($content);
157     }
158 }
159
160
161 // -----
162
163
164
165 /**
166 * Compressing and downloading test data and descriptions of an assignment to the browser
167 */
168
169 public function downloadtestsdesc($assignment_id = FALSE)
170 {
171     if ($assignment_id === FALSE)
172         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();

```

```

173|     if ( $this->user->level <= 1 ) // permission denied
174|         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
175|
176|     $assignment = $this->assignment_model->assignment_info($assignment_id);
177|
178|     $number_of_problems = $assignment['problems'];
179|
180|     $root_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
181|     "/$assignment[$assignment_id]";
182|     $zipname = "assignment{$assignment_id}_tests_desc_.date('Y-m-d_H-i', shj_now()).zip";
183|     $this->zip = new \ZipArchive();
184|     $this->zip->open($zipname, ZipArchive::CREATE);
185|     for ($i=1 ; $i<=$number_of_problems ; $i++)
186|     {
187|
188|         $path = "$root_path/p{$i}/in";
189|         $options = ['add_path' => "{$i}/in/", 'remove_all_path' => TRUE];
190|         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
191|
192|         $path = "$root_path/p{$i}/out";
193|         $options = ['add_path' => "{$i}/out/", 'remove_all_path' => TRUE];
194|         $this->zip->addGlob($path.'/*.{txt}', GLOB_BRACE, $options);
195|
196|         $path = "$root_path/p{$i}/tester.cpp";
197|         if (file_exists($path))
198|             $this->zip->addFile($path,"{$i}/tester.cpp");
199|
200|         $pdf_files = glob("$root_path/p{$i}/*.pdf");
201|         if ($pdf_files)
202|         {
203|             $path = $pdf_files[0];
204|             $this->zip->addFile($path,"{$i}/".shj_basename($path));
205|         }
206|
207|         $path = "$root_path/p{$i}/desc.html";
208|         if (file_exists($path))
209|             $this->zip->addFile($path,"{$i}/desc.html");
210|
211|         $path = "$root_path/p{$i}/desc.md";
212|         if (file_exists($path))
213|             $this->zip->addFile($path,"{$i}/desc.md");
214|
215|         $pdf_files = glob("$root_path/*.*");
216|         if ($pdf_files)
217|         {
218|             $path = $pdf_files[0];
219|             $this->zip->addFile($path,shj_basename($path));
220|         }
221|
222|     $this->zip->close();
223|
224|     header('Content-Type:application/zip');
225|     header('Content-disposition:attachment;filename=' . $zipname);
226|     header('Content-Length:' . filesize($zipname));
227|     readfile($zipname);
228|     // $this->zip->download("assignment{$assignment_id}_tests_desc_.date('Y-m-d_H-i', shj_now()).zip");
229}
230
231// -----
232
233
234/*
235 * Compressing and downloading final codes of an assignment to the browser
236 */
237
238 public function download_submissions($type = FALSE, $assignment_id = FALSE)
239{
240    if ($type !== 'by_user' && $type !== 'by_problem')
241        throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
242    if ($assignment_id === FALSE || ! is_numeric($assignment_id))
243        throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
244    if ( $this->user->level == 0 ) // permission denied
245        throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
246
247    $this->submit_model = new SubmitModel();
248    $items = $this->submit_model->get_final_submissions($assignment_id, $this->user->level, $this->user->username);
249    $zipname = "assignment{$assignment_id}_submissions_{$type}_date('Y-m-d_H-i', shj_now()).zip";
250
251    $this->zip = new \ZipArchive();
252    $this->zip->open($zipname, ZipArchive::CREATE);
253    $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
254
255    foreach ($items as $item)
256    {
257        $file_path = $assignments_root.
258            "/$assignment[$item['assignment']]/$item['problem']/{$item['username']}/{$item['file_name']}.".
259            filetype_to_extension($item['file_type']);
260        if ( ! file_exists($file_path) )
261            continue;
262        $file = file_get_contents($file_path);
263
264        if ($type === 'by_user')
265            $this->zip->addFile($file_path, "{$item['username']}/$item['problem'].".filetype_to_extension($item['file_type']));
266        elseif ($type === 'by_problem')
267            $this->zip->addFile($file_path, "problem_{$item['problem']}->{$item['username']}.".
268                filetype_to_extension($item['file_type']));
269    }
269 $this->zip->close();

```

```

270     header('Content-Type:_application/zip');
271     header('Content-disposition:_attachment;_filename='.$zipname);
272     header('Content-Length:_' . filesize($zipname));
273     readfile($zipname);
274 }
275
276
277 // -----
278
279
280 /**
281 * Delete assignment
282 */
283 public function delete($assignment_id = FALSE)
284 {
285     if ($assignment_id === FALSE)
286         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
287     if ($this->user->level <= 1) // permission denied
288         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
289
290     $assignment = $this->assignment_model->assignment_info($assignment_id);
291
292     if ($assignment['id'] === 0)
293         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
294
295     if ($this->request->getPost('delete') === 'delete')
296     {
297         $this->assignment_model->delete_assignment($assignment_id);
298         return redirect()->to('assignments');
299     }
300
301     $data = array(
302         'all_assignments' => $this->assignment_model->all_assignments(),
303         'id' => $assignment_id,
304         'name' => $assignment['name'],
305         'selected' => 'assignments',
306         'icon' => 'fa-times',
307         'title' => 'Delete Assignments',
308         'user' => $this->user,
309         'finish_time' => $this->user->selected_assignment['finish_time'],
310         'extra_time' => $this->user->selected_assignment['extra_time'],
311
312     );
313
314     return view('pages/admin/delete_assignment', $data);
315 }
316
317
318 // -----
319
320
321 /**
322 * This method gets inputs from user for adding/editing assignment
323 */
324 public function add()
325 {
326
327     if ($this->user->level <= 1) // permission denied
328         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
329
330
331     if ($this->request->is('post')){
332         if ($this->add()) // add/edit assignment
333         {
334             //if ( ! $this->edit) // if adding assignment (not editing)
335             //{
336                 // goto Assignments page
337                 return $this->index();
338             //}
339         }
340     }
341
342     $data = array(
343         'all_assignments' => $this->assignment_model->all_assignments(),
344         'messages' => $this->messages,
345         'edit' => $this->edit,
346         'default_late_rule' => $this->settings_model->get_setting('default_late_rule'),
347         'selected' => 'assignments',
348         'icon' => 'fa-edit',
349         'title' => 'Assignments',
350         'user' => $this->user,
351         'finish_time' => $this->user->selected_assignment['finish_time'],
352         'extra_time' => $this->user->selected_assignment['extra_time'],
353         'twig' => $this->twig,
354         'validationError' => $this->validation,
355
356     );
357
358     if ($this->edit)
359     {
360         $data['edit_assignment'] = $this->assignment_model->assignment_info($this->edit_assignment);
361         if ($data['edit_assignment']['id'] === 0)
362             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
363         $data['problems'] = $this->assignment_model->all_problems($this->edit_assignment);
364     }
365     else
366     {
367
368

```

```

369     $names = $this->request->getPost('name');
370     if ($names === NULL)
371         $data['problems'] = array(
372             array(
373                 'id' => 1,
374                 'name' => 'Problem_',
375                 'score' => 100,
376                 'c_time_limit' => 500,
377                 'python_time_limit' => 1500,
378                 'java_time_limit' => 2000,
379                 'memory_limit' => 50000,
380                 'allowed_languages' => 'C,C++,Python_2,Python_3,Java',
381                 'diff_cmd' => 'diff',
382                 'diff_arg' => '-bB',
383                 'is_upload_only' => 0
384             )
385         );
386     else
387     {
388         $names = $this->request->getPost('name');
389         $scores = $this->request->getPost('score');
390         $c_tl = $this->request->getPost('c_time_limit');
391         $py_tl = $this->request->getPost('python_time_limit');
392         $java_tl = $this->request->getPost('java_time_limit');
393         $ml = $this->request->getPost('memory_limit');
394         $ft = $this->request->getPost('languages');
395         $dc = $this->request->getPost('diff_cmd');
396         $da = $this->request->getPost('diff_arg');
397         $data[ 'problems' ] = array();
398         $uo = $this->request->getPost('is_upload_only');
399         if ($uo === NULL)
400             $uo = array();
401         for ($i=0; $i<count($names); $i++){
402             array_push($data['problems'], array(
403                 'id' => $i+1,
404                 'name' => $names[$i],
405                 'score' => $scores[$i],
406                 'c_time_limit' => $c_tl[$i],
407                 'python_time_limit' => $py_tl[$i],
408                 'java_time_limit' => $java_tl[$i],
409                 'memory_limit' => $ml[$i],
410                 'allowed_languages' => $ft[$i],
411                 'diff_cmd' => $dc[$i],
412                 'diff_arg' => $da[$i],
413                 'is_upload_only' => in_array($i+1,$uo)?1:0,
414             ));
415         }
416     }
417 }
418
419 return view('pages/admin/add_assignment', $data);
420 }
421
422 // -----
423
424
425 /**
426 * Add/Edit assignment
427 */
428
429 private function _add()
430 {
431     // Check permission
432     if ($this->user->level <= 1) // permission denied
433         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
434
435     $this->validation->setRule('assignment_name', 'assignment_name', 'required|max_length[50]');
436     $this->validation->setRule('start_time', 'start_time', 'required');
437     $this->validation->setRule('finish_time', 'finish_time', 'required');
438     $this->validation->setRule('extra_time', 'extra_time', 'required');
439     $this->validation->setRule('participants', 'participants', 'string');
440     $this->validation->setRule('late_rule', 'coefficient_rule', 'required');
441     $this->validation->setRule('name.*', 'problem_name', 'required|max_length[50]');
442     $this->validation->setRule('score.*', 'problem_score', 'required|integer');
443     $this->validation->setRule('c_time_limit.*', 'C/C++_time_limit', 'required|integer');
444     $this->validation->setRule('python_time_limit.*', 'python_time_limit', 'required|integer');
445     $this->validation->setRule('java_time_limit.*', 'java_time_limit', 'required|integer');
446     $this->validation->setRule('memory_limit.*', 'memory_limit', 'required|integer');
447     $this->validation->setRule('languages.*', 'languages', 'required');
448     $this->validation->setRule('diff_cmd.*', 'diff_command', 'required');
449     $this->validation->setRule('diff_arg.*', 'diff_argument', 'required');
450
451     // Validate input data
452     if ( !$this->validation->withRequest($this->request)->run() ){
453         return FALSE;
454     }
455
456     // Preparing variables
457
458     if ($this->edit)
459         $the_id = $this->edit_assignment;
460     else
461         $the_id = $this->assignment_model->new_assignment_id();
462
463     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
464     $assignment_dir = "$assignments_root/assignment_{$the_id}";
465
466     // Adding/Editing assignment in database
467     if ( !$this->assignment_model->add_assignment($the_id, $this->edit))

```

```

468{
469    $this->messages[] = array(
470        'type' => 'error',
471        'text' => 'Error'.($this->edit?'updating':'adding').'_assignment.'
472    );
473    return FALSE;
474}
475
476$this->messages[] = array(
477    'type' => 'success',
478    'text' => 'Assignment'.($this->edit?'updated':'added').'_successfully.'
479);
480
481// Create assignment directory
482if( !file_exists($assignment_dir) ){
483    mkdir($assignment_dir, 0700, true);
484}
485
486// Upload Tests (zip file)
487$files = glob($assignments_root.'/*.zip');
488foreach($files as $file){
489    if(is_file($file)){
490        // Delete the given file
491        unlink($file);
492    }
493}
494$zip_uploaded = $this->request->getFile('tests_desc');
495if( $_FILES['tests_desc']['error'] === UPLOAD_ERR_NO_FILE ){
496    $this->messages[] = array(
497        'type' => 'notice',
498        'text' => "Notice: You did not upload any zip file for tests. If needed, upload by editing assignment."
499    );
500}
501elseif( $zip_uploaded->getExtension() != "zip" ){
502    $this->messages[] = array(
503        'type' => 'error',
504        'text' => "Error: Error uploading tests zip file: The filetype you are attempting to upload is not allowed."
505    );
506}
507else{
508    $zip_uploaded->move($assignments_root);
509    $this->messages[] = array(
510        'type' => 'success',
511        'text' => "Tests(zip file) uploaded successfully."
512    );
513}
514
515// Upload PDF File of Assignment
516$old_pdf_files = glob("$assignment_dir/*.pdf");
517$pdf_uploaded = $this->request->getFile('pdf');
518if( $_FILES['pdf']['error'] === UPLOAD_ERR_NO_FILE){
519    $this->messages[] = array(
520        'type' => 'notice',
521        'text' => "Notice: You did not upload any pdf file for assignment. If needed, upload by editing assignment."
522    );
523}
524elseif( $pdf_uploaded->getExtension() != "pdf" ){
525    $this->messages[] = array(
526        'type' => 'error',
527        'text' => "Error: Error uploading pdf file of assignment: The filetype you are attempting to upload is not allowed."
528    );
529}
530else{
531    $pdf_uploaded->move("$assignment_dir");
532    {
533        foreach($old_pdf_files as $old_name){
534            unlink($old_name);
535        }
536        $this->messages[] = array(
537            'type' => 'success',
538            'text' => 'PDF file uploaded successfully.'
539        );
540    }
541}
542
543
544// Extract Tests (zip file)
545if( $zip_uploaded->getClientOriginalExtension() == "zip" ){ // if zip file is uploaded
546    $this->unzip = new ZipArchive();
547    // Create a temp directory
548    $tmp_dir_name = "shj_tmp_directory";
549    $tmp_dir = "$assignments_root/$tmp_dir_name";
550    shell_exec("rm -rf $tmp_dir; mkdir $tmp_dir");
551
552    // Extract new test cases and descriptions in temp directory
553    $this->unzip->open("$assignments_root/".$zip_uploaded->getName());
554    $extract_result = $this->unzip->extractTo($tmp_dir);
555
556    // Remove the zip file
557    unlink("$assignments_root/".$zip_uploaded->getName());
558
559    if( $extract_result )
560    {
561        // Remove previous test cases and descriptions
562        shell_exec("cd $assignment_dir;" .
563                  ". rm -rf */in; rm -rf */out; rm -f */tester.cpp; rm -f */tester.executable;" .
564                  ". rm -f */desc.html; rm -f */desc.md; rm -f */*.pdf;");
565        if( glob("$tmp_dir/*.pdf") )
566    }

```

```

566         shell_exec("cd $assignment_dir; rm -f *.pdf");
567         // Copy new test cases from temp dir
568         shell_exec("cd $assignments_root; cp -R $tmp_dir_name/* $assignment_{$the_id};");
569         $this->messages[] = array(
570             'type' => 'success',
571             'text' => 'Tests_(zip_file)_extracted_successfully.'
572         );
573     }
574     else
575     {
576         $this->messages[] = array(
577             'type' => 'error',
578             'text' => 'Error:_Error_extracting_zip_archive.'
579         );
580         $this->messages[] = array(
581             'type' => 'error',
582             'text' => "_Zip_Extraction_Error:_" . $this->unzip->getStatusString(),
583         );
584     }
585     // Remove temp directory
586     shell_exec("rm -rf $tmp_dir");
587 }
588
589 // Create problem directories and parsing markdown files
590
591 for ($i=1; $i <= $this->request->getPost('number_of_problems'); $i++)
592 {
593     if ( ! file_exists("$assignment_dir/p{$i}")
594         mkdir("$assignment_dir/p{$i}", 0700);
595     elseif (file_exists("$assignment_dir/p{$i}/desc.md"))
596     {
597         $html = $this->parsedown->parse(file_get_contents("$assignment_dir/p{$i}/desc.md"));
598         file_put_contents("$assignment_dir/p{$i}/desc.html", $html);
599     }
600 }
601
602 return TRUE;
603 }
604
605
606 // -----
607
608
609 public function edit($assignment_id)
610 {
611
612     if ($this->user->level <= 1) // permission denied
613         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
614
615     $this->edit_assignment = $assignment_id;
616     $this->edit = TRUE;
617
618     // redirect to add function
619     return $this->add();
620 }
621
622
623
624 // -----
625
626
627 /**
628 * Check PDF File Availability
629 */
630
631 public function pdfCheck($assignment_id, $problem_id = NULL)
632 {
633     $finishtime = strtotime($this->assignment_model->assignment_info($assignment_id)['finish_time']);
634     $starttime = strtotime($this->assignment_model->assignment_info($assignment_id)['start_time']);
635     $extratime = $this->assignment_model->assignment_info($assignment_id)['extra_time'];
636
637     // Find pdf file
638     if ($problem_id === NULL || $problem_id === "null")
639         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/*.pdf";
640     else
641         $pattern = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/p{$problem_id}/*.pdf";
642     $pdf_files = glob($pattern);
643
644     if ( ! $pdf_files )
645         $response = json_encode(array('status'=>FALSE));
646     elseif ( !$this->assignment_model->assignment_info($assignment_id)['open'] )
647         $response = json_encode(array('status'=>FALSE));
648     elseif ( ! $this->assignment_model->is_participant($this->assignment_model->assignment_info($assignment_id)['participants'],
649                 $this->user->username) )
650         $response = json_encode(array('status'=>FALSE));
651     elseif ( shj_now() > $finishtime + $extratime )
652         $response = json_encode(array('status'=>FALSE));
653     elseif ( shj_now() < $starttime )
654         $response = json_encode(array('status'=>FALSE));
655     else
656         $response = json_encode(array('status'=>TRUE));
657
658     echo $response;
659 }

```

Kode A.2: BaseController.php

```

1 <?php
2
3 namespace App\Controllers;
4
5 use App\Libraries\Parsedown;
6 use App\Libraries\Unzip;
7 use App\Libraries\Twig;
8 use CodeIgniter\Controller;
9 use CodeIgniter\HTTP\CLIRequest;
10 use CodeIgniter\HTTP\IncomingRequest;
11 use CodeIgniter\HTTP\RequestInterface;
12 use CodeIgniter\HTTP\ResponseInterface;
13 use Psr\Log\LoggerInterface;
14
15 /**
16  * Class BaseController
17  *
18  * BaseController provides a convenient place for loading components
19  * and performing functions that are needed by all your controllers.
20  * Extend this class in any new controllers:
21  *     class Home extends BaseController
22  *
23  * For security be sure to declare any new methods as protected or private.
24  */
25 abstract class BaseController extends Controller
26 {
27     /**
28      * Instance of the main Request object.
29      *
30      * @var CLIRequest|IncomingRequest
31      */
32     protected $request;
33
34     /**
35      * An array of helpers to be loaded automatically upon
36      * class instantiation. These helpers will be available
37      * to all other controllers that extend BaseController.
38      *
39      * @var array
40      */
41     protected $helpers = ['text','url','shj_helper','form','cookie','string','filesystem'];
42
43     /**
44      * Be sure to declare properties for any property fetch you initialized.
45      * The creation of dynamic property is deprecated in PHP 8.2.
46      */
47     protected $parsedown;
48     protected $twig;
49
50     /**
51      * Constructor.
52      */
53     public function initController(RequestInterface $request, ResponseInterface $response, LoggerInterface $logger)
54     {
55         // Do Not Edit This Line
56         parent::initController($request, $response, $logger);
57
58         // Preload any models, libraries, etc, here.
59
60         // E.g.: $this->session = \Config\Services::session();
61         $appPaths = new \Config.Paths();
62         $appViewPaths = $appPaths->viewDirectory;
63         $this->parsedown = new Parsedown();
64         $this->twig = new Twig();
65     }
66 }

```

Kode A.3: Dashboard.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Dashboard.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7
8 namespace App\Controllers;
9
10 use App\Controllers\BaseController;
11 use App\Models\AssignmentModel;
12 use App\Models\NotificationsModel;
13 use App\Models\SettingsModel;
14 use App\Models\User;
15 use App\Models\UserModel;
16 use Exception;
17
18 class Dashboard extends BaseController
19 {
20     protected $db;
21     protected $assignment_model;
22     protected $notifications_model;
23     protected $settings_model;
24     protected $user;
25     protected $session;
26     protected $user_model;
27

```

```

28|     public function __construct()
29|     {
30|         $this->session = session();
31|         $this->assignment_model = new AssignmentModel();
32|         $this->settings_model = new SettingsModel();
33|         $this->user = new User();
34|         $this->user_model = new UserModel();
35|         $this->notifications_model = new NotificationsModel();
36|     }
37|
38|     // -----
39|
40|
41|     public function index()
42|     {
43|         $data = array(
44|             'all_assignments' =>$this->assignment_model->all_assignments(),
45|             'week_start'=>$this->settings_model->get_setting('week_start'),
46|             'wp'=>$this->user->get_widget_positions(),
47|             'notifications' => $this->notifications_model->get_latest_notifications(),
48|             'selected' => 'dashboard',
49|             'user' => $this->user,
50|             'finish_time' => $this->user->selected_assignment['finish_time'],
51|             'extra_time' => $this->user->selected_assignment['extra_time'],
52|         );
53|
54|
55|         // detecting errors:
56|         $data['errors'] = array();
57|         if($this->user->level == 3){
58|             $path = $this->settings_model->get_setting('assignments_root');
59|             if ( ! file_exists($path))
60|                 array_push($data['errors'], 'The_path_to_folder_"assignments"_is_not_set_correctly._Move_this_folder_somewhere_not_
61|                                         _publicly_accessible,_and_set_its_full_path_in_Settings.');
62|             elseif ( ! is_writable($path))
63|                 array_push($data['errors'], 'The_folder_<code>"'$path.'</code>_is_not_writable_by_PHP._Make_it_writable._But_
64|                                         make_sure_that_this_folder_is_only_accessible_by_you._Codes_will_be_saved_in_this_folder!');
65|
66|             $path = $this->settings_model->get_setting('tester_path');
67|             if ( ! file_exists($path))
68|                 array_push($data['errors'], 'The_path_to_folder_"tester"_is_not_set_correctly._Move_this_folder_somewhere_not_
69|                                         _publicly_accessible,_and_set_its_full_path_in_Settings.');
70|             elseif ( ! is_writable($path))
71|                 array_push($data['errors'], 'The_folder_<code>"'$path.'</code>_is_not_writable_by_PHP._Make_it_writable._But_
72|                                         make_sure_that_this_folder_is_only_accessible_by_you.');
73|         }
74|
75|         // -----
76|
77|         /**
78|          * Used by ajax request, for saving the user's Dashboard widget positions
79|          */
80|
81|         public function widget_positions()
82|         {
83|             if ( ! $this->request->isAJAX() )
84|                 throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
85|             if ($this->request->getPost('positions') !== NULL)
86|                 $this->user->save_widget_positions($this->request->getPost('positions'));
87|         }
88|
89|     }

```

Kode A.4: Halloffame.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Halloffame.php
5  * @author Stillmen Vallian <stillmen.v@gmail.com>
6  */
7 namespace App\Controllers;
8
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\HofModel;
12 use App\Models\User;
13
14 class Halloffame extends BaseController
15 {
16     protected $hof_model;
17     protected $user;
18     protected $assignment_model;
19
20     public function __construct()
21     {
22         $this->hof_model = new HofModel();
23         $this->user = new User();
24         $this->assignment_model = new AssignmentModel();
25     }
26
27
28
29

```

```

30 // -----
31
32
33
34
35     public function index()
36     {
37         $data = array(
38             'hof' => $this->hof_model->get_all_final_submission(),
39             'user' => $this->user,
40             'selected' => 'halloffame',
41             'finish_time' => $this->user->selected_assignment['finish_time'],
42             'extra_time' => $this->user->selected_assignment['extra_time'],
43             'all_assignments' => $this->assignment_model->all_assignments(),
44         );
45         return view ('pages/halloffame', $data);
46     }
47
48
49
50
51 // -----
52
53
54
55 /**
56 * Controller for shows the details of the score
57 * Called by ajax request
58 */
59     public function hof_details()
60     {
61         if ( ! $this->request->isAjax() )
62             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
63         $username = $this->request->getVar('username');
64
65         $json_result = $this->hof_model->get_all_user_assignments($username);
66
67         $this->response->setHeader('Content-Type', 'application/json; charset=utf-8');
68         echo json_encode($json_result);
69     }
70 }
```

Kode A.5: Install.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4 * @file Install.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7 namespace App\Controllers;
8
9 use App\Controllers\BaseController;
10 use App\Models\UserModel;
11 use CodeIgniter\Database\RawSql;
12 use Exception;
13
14 class Install extends BaseController
15 {
16
17     protected $db;
18     protected $dbforge;
19     protected $user_model;
20     protected $validation;
21     protected $validationError;
22
23     public function __construct()
24     {
25         $this->db = db_connect();
26         $this->dbforge = \Config\Database::forge();
27         $this->user_model = new UserModel();
28         $this->validation = \Config\Services::validation();
29     }
30
31
32 // -----
33
34     public function index()
35     {
36
37         try{
38             if ($this->db->tableExists('sessions'))
39                 throw new \Exception('SharIF_Judge_is_already_installed.');
40         } catch (\Exception $e){
41             exit($e->getMessage());
42         }
43
44         $this->validation->setRule('username', 'username', 'required|min_length[3]|max_length[20]|alpha_numeric');
45         $this->validation->setRule('email', 'email', 'required|max_length[40]|valid_email');
46         $this->validation->setRule('password', 'password', 'required|min_length[6]|max_length[200]');
47         $this->validation->setRule('password_again', 'password_confirmation', 'required|matches[password]');
48
49         $data['installed'] = FALSE;
50         $data['title'] = 'Installation';
51         $data['validationError'] = $this->validation;
52         $data['key_changed'] = FALSE;
53
54         if($this->request->is('post')){
```

```

55|     if ($this->validation->withRequest($this->request)->run()) {
56|         $DATETIME = 'DATETIME';
57|         if ($this->db->DBDriver === 'postgre')
58|             $DATETIME = 'TIMESTAMP';
59|
60|         // Use InnoDB engine for MySql database
61|         if ($this->db->DBDriver === 'mysql' || $this->db->DBDriver === 'mysqli')
62|             $this->db->query('SET_storage_engine=InnoDB;');
63|
64|         // Creating Tables:
65|         // sessions, submissions, assignments, notifications, problems, queue, scoreboard, settings, users
66|
67|
68|         // create table 'sessions'
69|         $fields = [
70|             'id' => ['type' => 'VARCHAR', 'constraint' => 128, 'null' => false],
71|             'ip_address' => ['type' => 'VARCHAR', 'constraint' => 45, 'null' => false],
72|             'timestamp' => ['type' => 'TIMESTAMP', 'default' => new RawSql('CURRENT_TIMESTAMP'), 'null' => false],
73|             'data' => ['type' => 'blob', 'null' => false],
74|         ];
75|         $this->dbforge->addField($fields);
76|         $this->dbforge->addKey('id', TRUE); // PRIMARY KEY
77|         try{
78|             if ( ! $this->dbforge->createTable('sessions', TRUE))
79|                 throw new \Exception("Error_creating_database_table_". $this->db->prefixTable('sessions'));
80|         }catch(Exception $e){
81|             exit($e->getMessage());
82|         }
83|
84|
85|         // create table 'submissions'
86|         $fields = [
87|             'submit_id' => ['type' => 'INT', 'constraint' => 11, 'unsigned' => TRUE],
88|             'username' => ['type' => 'VARCHAR', 'constraint' => 20],
89|             'assignment' => ['type' => 'SMALLINT', 'constraint' => 4, 'unsigned' => TRUE],
90|             'problem' => ['type' => 'SMALLINT', 'constraint' => 4, 'unsigned' => TRUE],
91|             'is_final' => ['type' => 'TINYINT', 'constraint' => 1, 'default' => 0],
92|             'time' => ['type' => $DATETIME],
93|             'status' => ['type' => 'VARCHAR', 'constraint' => 100],
94|             'pre_score' => ['type' => 'INT', 'constraint' => 11],
95|             'coefficient' => ['type' => 'VARCHAR', 'constraint' => 6],
96|             'file_name' => ['type' => 'VARCHAR', 'constraint' => 30],
97|             'main_file_name'=> ['type' => 'VARCHAR', 'constraint' => 30],
98|             'file_type' => ['type' => 'VARCHAR', 'constraint' => 6],
99|         ];
100|        $this->dbforge->addField($fields);
101|        $this->dbforge->addKey(['assignment', 'submit_id']);
102|        try{
103|            if ( ! $this->dbforge->createTable('submissions', TRUE))
104|                throw new \Exception("Error_creating_database_table_". $this->db->prefixTable('submissions'));
105|        }catch(Exception $e){
106|            exit($e->getMessage());
107|        }
108|
109|
110|        // create table 'assignments'
111|        $fields = [
112|            'id' => ['type' => 'INT', 'constraint' => 11, 'unsigned' => TRUE, 'auto_increment' => TRUE],
113|            'name' => ['type' => 'VARCHAR', 'constraint' => 50, 'default' => ''],
114|            'problems' => ['type' => 'SMALLINT', 'constraint' => 4, 'unsigned' => TRUE],
115|            'total_submits' => ['type' => 'INT', 'constraint' => 11, 'unsigned' => TRUE],
116|            'open' => ['type' => 'TINYINT', 'constraint' => 1],
117|            'scoreboard' => ['type' => 'TINYINT', 'constraint' => 1],
118|            'javaexceptions'=> ['type' => 'TINYINT', 'constraint' => 1],
119|            'description' => ['type' => 'TEXT'],
120|            'start_time' => ['type' => $DATETIME],
121|            'finish_time' => ['type' => $DATETIME],
122|            'extra_time' => ['type' => 'INT', 'constraint' => 11],
123|            'late_rule' => ['type' => 'TEXT'],
124|            'participants' => ['type' => 'TEXT'],
125|            'moss_update' => ['type' => 'VARCHAR', 'constraint' => 30, 'default' => 'Never'],
126|            'archived_assignment' => ['type' => 'TINYINT', 'constraint' => 1],
127|        ];
128|        $this->dbforge->addField($fields);
129|        $this->dbforge->addKey('id', TRUE); // PRIMARY KEY
130|        try{
131|            if ( ! $this->dbforge->createTable('assignments', TRUE))
132|                throw new \Exception("Error_creating_database_table_". $this->db->prefixTable('assignments'));
133|        }catch(Exception $e){
134|            exit($e->getMessage());
135|        }
136|
137|
138|        // create table 'notifications'
139|        $fields = [
140|            'id' => ['type' => 'INT', 'constraint' => 11, 'unsigned' => TRUE, 'auto_increment' => TRUE],
141|            'title' => ['type' => 'VARCHAR', 'constraint' => 200, 'default' => ''],
142|            'text' => ['type' => 'TEXT'],
143|            'time' => ['type' => $DATETIME],
144|        ];
145|        $this->dbforge->addField($fields);
146|        $this->dbforge->addKey('id', TRUE); // PRIMARY KEY
147|        try{
148|            if ( ! $this->dbforge->createTable('notifications', TRUE))
149|                throw new \Exception("Error_creating_database_table_". $this->db->prefixTable('notifications'));
150|        }catch(Exception $e){
151|            exit($e->getMessage());
152|        }
153|

```



```

245     n_\$coefficient_=>0;"],  

246     ['shj_key' => 'enable_easysandbox', 'shj_value' => '1'],  

247     ['shj_key' => 'enable_c_shield', 'shj_value' => '1'],  

248     ['shj_key' => 'enable_cpp_shield', 'shj_value' => '1'],  

249     ['shj_key' => 'enable_py2_shield', 'shj_value' => '1'],  

250     ['shj_key' => 'enable_py3_shield', 'shj_value' => '1'],  

251     ['shj_key' => 'enable_java_policy', 'shj_value' => '1'],  

252     ['shj_key' => 'enable_log', 'shj_value' => '1'],  

253     ['shj_key' => 'submit_penalty', 'shj_value' => '300'],  

254     ['shj_key' => 'enable_registration', 'shj_value' => '0'],  

255     ['shj_key' => 'registration_code', 'shj_value' => '0'],  

256     ['shj_key' => 'mail_from', 'shj_value' => 'no-reply+shj@labftis.net'],  

257     ['shj_key' => 'mail_from_name', 'shj_value' => "Judge from FTIS Administrator"],  

258     ['shj_key' => 'reset_password_mail', 'shj_value' => "<p>\nSomeone requested a password reset for your  
SharIF Judge account at {SITE_URL}.</p>\nTo change your password, visit this link:<br>\n<a href=\"{RESET_LINK}\">Reset Password</a></p>\nThe link is valid for {VALID_TIME}. If you don't want to change your password, just ignore this email.</p>"],  

259     ['shj_key' => 'add_user_email', 'shj_value' => "<p>\nHello! You are registered in SharIF Judge at {  
SITE_URL} as {ROLE}.</p>\n<p>\nYour username: {USERNAME}</p>\n<p>\nYour password: {PASSWORD}</p>\n<p>\nYou can log in at <a href=\"{LOGIN_URL}\">{LOGIN_URL}</a></p>"],  

260     ['shj_key' => 'moss_userid', 'shj_value' => ''],  

261     ['shj_key' => 'results_per_page_all', 'shj_value' => '40'],  

262     ['shj_key' => 'results_per_page_final', 'shj_value' => '80'],  

263     ['shj_key' => 'week_start', 'shj_value' => '0'],  

264     ['shj_key' => 'lock_student_display_name', 'shj_value' => '0'],  

265   ];  

266   $builder = $this->db->table('settings');  

267   try{  

268     if ( ! $builder->insertBatch($result))  

269       throw new \Exception("Error adding data to table ".$this->db->prefixTable('settings'));  

270   }catch(\Exception $e){  

271     exit($e->getMessage());  

272   }  

273  

274   // create table 'users'  

275   $fields = [  

276     'id' => ['type' => 'INT', 'constraint' => 11, 'unsigned' => TRUE, 'auto_increment' => TRUE],  

277     'username' => ['type' => 'VARCHAR', 'constraint' => 20],  

278     'password' => ['type' => 'VARCHAR', 'constraint' => 100],  

279     'display_name' => ['type' => 'VARCHAR', 'constraint' => 40, 'default' => ''],  

280     'email' => ['type' => 'VARCHAR', 'constraint' => 40],  

281     'role' => ['type' => 'VARCHAR', 'constraint' => 20],  

282     'passchange_key' => ['type' => 'VARCHAR', 'constraint' => 60, 'default' => ''],  

283     'passchange_time' => ['type' => DATETIME, 'null' => TRUE],  

284     'first_login_time' => ['type' => DATETIME, 'null' => TRUE],  

285     'last_login_time' => ['type' => DATETIME, 'null' => TRUE],  

286     'selected_assignment' => ['type' => 'SMALLINT', 'constraint' => 4, 'unsigned' => TRUE, 'default' => 0],  

287     'dashboard_widget_positions' => ['type' => 'VARCHAR', 'constraint' => 500, 'default' => ''],  

288   ];  

289   $this->dbforge->addField($fields);  

290   $this->dbforge->addKey('id', TRUE); // PRIMARY KEY  

291   $this->dbforge->addKey('username'); // @todo is this needed?  

292   try{  

293     if ( ! $this->dbforge->createTable('users', TRUE))  

294       throw new \Exception("Error creating database table ".$this->db->prefixTable('users'));  

295   }catch(\Exception $e){  

296     exit($e->getMessage());  

297   }  

298  

299   // create table 'logins'  

300   $fields = [  

301     'login_id' => ['type' => 'INT', 'constraint' => 11, 'unsigned' => TRUE, 'auto_increment' => TRUE],  

302     'username' => ['type' => 'VARCHAR', 'constraint' => 20],  

303     'ip_address' => ['type' => 'VARCHAR', 'constraint' => 15],  

304     'timestamp' => ['type' => 'TIMESTAMP'],  

305     'last_24h_login_id' => ['type' => 'INT', 'constraint' => 11, 'null' => TRUE],  

306   ];  

307   $this->dbforge->addField($fields);  

308   $this->dbforge->addKey('login_id', TRUE); // PRIMARY KEY  

309   try{  

310     if ( ! $this->dbforge->createTable('logins', TRUE))  

311       throw new \Exception("Error creating database table ".$this->db->prefixTable('logins'));  

312   }catch(\Exception $e){  

313     exit($e->getMessage());  

314   }  

315  

316   // add admin user  

317   $this->user_model->add_user(  

318     $this->request->getPost('username'),  

319     $this->request->getPost('email'),  

320     'Admin',  

321     $this->request->getPost('password'),  

322     'admin',  

323   );  

324   $data['installed'] = TRUE;  

325 }  

326  

327 // Using a random string as encryption key  

328 $config_path = rtrim(APPPATH, '/') . '/Config/Encryption.php';  

329 $config_content = file_get_contents($config_path);  

330 $random_key = random_string('alnum', 32);  

331 $res = @file_put_contents($config_path, str_replace(config('Encryption')->key, $random_key, $config_content));  

332 if ($res === FALSE)  

333   $data['key_changed'] = FALSE;  

334 else  

335   $data['key_changed'] = TRUE;  

336 }  

337 $data['enc_key'] = config('Encryption')->key;

```

```

338     $data['random_key'] = random_string('alnum', 32);
339
340     return view('pages/admin/install', $data);
341
342 }
343

```

Kode A.6: Login.php

```

1 <?php
2 namespace App\Controllers;
3
4 use App\Controllers\BaseController;
5 use App\Models\UserModel;
6 use App\Models\LogsModel;
7 use App\Models\SettingsModel;
8
9 /**
10  * SharIF Judge online judge
11  * @file Login.php
12  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
13 */
14
15
16 class Login extends BaseController
17 {
18     protected $settings_model;
19     protected $validation;
20     protected $session;
21     protected $user_model;
22     protected $logs_model;
23
24     public function __construct()
25     {
26         $this->settings_model = new SettingsModel();
27         $this->user_model = new UserModel();
28         $this->logs_model = new LogsModel();
29         $this->validation = \Config\Services::validation();
30         $this->session = session();
31     }
32
33
34 // -----
35
36
37 /**
38  * checks whether the entered registration code is correct or not
39  *
40  */
41 public function _registration_code($code){
42     $rc = $this->settings_model->get_setting('registration_code');
43     if ($rc == '0')
44         return TRUE;
45     if ($rc == $code)
46         return TRUE;
47     return FALSE;
48 }
49
50
51 // -----
52
53
54 /**
55  * Login
56  */
57 public function index()
58 {
59     if ($this->session->get('logged_in')) // if logged in
60         return redirect()->to('/');
61
62     $this->validation->setRules([
63         'username' => ['label' => 'Username', 'rules' => 'required|min_length[3]|max_length[20]|alpha_numeric',['required' => 'username.required']],
64         'password' => ['label' => 'Password', 'rules' => 'required|min_length[6]|max_length[200]']
65     ]);
66
67     $data = [
68         'error' => FALSE,
69         'registration_enabled' => $this->settings_model->get_setting('enable_registration'),
70         'title' => 'Login',
71         'validationError' => $this->validation
72     ];
73     if($this->request->is('post')){
74         if($this->validation->withRequest($this->request)->run()){
75             $username = $this->request->getPost('username');
76             $password = $this->request->getPost('password');
77             if($this->user_model->validate_user($username, $password)){
78                 $ip_address = $this->request->getIpaddress();
79                 // setting the session and redirecting to dashboard:
80                 $login_data = array(
81                     'username' => $username,
82                     'logged_in' => TRUE
83                 );
84                 $this->session->set($login_data);
85                 $this->user_model->update_login_time($username);
86                 $this->logs_model->insert_to_logs($username,$ip_address);
87                 return redirect()->to('/');
88             }
89         }
90     }
91 }
92

```

```

89         else{
90             // for displaying error message in 'pages/authentication/login' view
91             $data['error'] = TRUE;
92             return view('pages/authentication/login', $data);
93         }
94     } else{
95         return view('pages/authentication/login', $data);
96     }
97 }
98
99
100 // -----
101
102
103 public function register()
104 {
105     if ($this->session->get('logged_in')) // if logged in
106         return redirect()->to('dashboard');
107     if (! $this->settings_model->get_setting('enable_registration'))
108         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Registration_is_Closed');
109     $this->validation->setRule('registration_code', 'registration_code', '_registration_code', array('_registration_code' => '
110         Invalid_registration_code'));
111     $this->validation->setRule('username', 'username', 'required|min_length[3]|max_length[20]|alpha_numeric|lowercase[username]
112         ]|is_unique[users.username]', array('is_unique' => 'This_username_already_exists.', 'alpha_numeric'));
113     $this->validation->setRule('email', 'email_address', 'required|max_length[40]|valid_email|lowercase[username]|is_unique[
114         users.email]', array('is_unique' => 'This_email_address_already_exists.'));
115     $this->validation->setRule('password', 'password', 'required|min_length[6]|max_length[200]');
116     $this->validation->setRule('password_again', 'password_confirmation', 'required|matches[password]');
117     $data = array(
118         'registration_code_required' => $this->settings_model->get_setting('registration_code')=='0'?FALSE:TRUE,
119         'title' => 'Registration',
120         'validationError' => $this->validation
121     );
122     if($this->request->is('post')){
123         if ($this->validation->withRequest($this->request)->run()){
124             $this->user_model->add_user(
125                 $this->request->getPost('username'),
126                 $this->request->getPost('email'),
127                 $this->request->getPost('displayname'),
128                 $this->request->getPost('password'),
129                 'student'
130             );
131             return view('pages/authentication/register_success', $data);
132         }
133     }
134     else{
135         return view('pages/authentication/register', $data);
136     }
137 }
138
139
140 // -----
141
142
143 /**
144  * Logs out and redirects to login page
145 */
146 public function logout()
147 {
148     $this->session->destroy();
149     return redirect()->to('/login');
150 }
151
152
153
154 // -----
155
156
157 public function lost()
158 {
159     if ($this->session->get('logged_in')) // if logged in
160         return redirect()->to('dashboard');
161     $this->validation->setRule('email', 'email', 'required|max_length[40]|lowercase[email]|valid_email');
162     $data = array(
163         'sent' => FALSE,
164         'title' => 'Reset_Password'
165     );
166     if ($this->validation->withRequest($this->request)->run())
167     {
168         $data['errors'] = $this->validation->getErrors();
169         $this->user_model->send_password_reset_mail($this->request->getPost('email'));
170         $data['sent'] = TRUE;
171     }
172
173     return view('pages/authentication/lost', $data);
174 }
175
176
177 // -----
178
179
180 public function reset($passchange_key = FALSE)
181 {
182     if ($passchange_key === FALSE)
183         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
184     $result = $this->user_model->passchange_is_valid($passchange_key);

```

```

185     if ($result !== TRUE)
186         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound($result);
187     $this->validation->setRule('password', 'password', 'required|min_length[6]|max_length[200]');
188     $this->validation->setRule('password_again', 'password_confirmation', 'required|matches[password]');
189     $data = array(
190         'key' => $passchange_key,
191         'result' => $result,
192         'reset' => FALSE,
193         'validationError' => $this->validation,
194         'title' => 'Reset_Password'
195     );
196     if($this->request->is('post')){
197         if ($this->validation->withRequest($this->request)->run()){
198             $this->user_model->reset_password($passchange_key, $this->request->getPost('password'));
199             $data['reset'] = TRUE;
200             return view('pages/authentication/reset_password', $data);
201         }else{
202             return view('pages/authentication/reset_password', $data);
203         }
204     }else{
205         return view('pages/authentication/reset_password', $data);
206     }
207 }
208
209
210
211 }

```

Kode A.7: Logs.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4 * @file Logs.php
5 * @author Stillmen Vallian <stillmen.v@gmail.com>
6 */
7 namespace App\Controllers;
8
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\LogsModel;
12 use App\Models\User;
13
14 class Logs extends BaseController
15 {
16     protected $session;
17     protected $user;
18     protected $logs_model;
19     protected $assignment_model;
20
21     public function __construct()
22     {
23         $this->session = session();
24         $this->logs_model = new LogsModel();
25         $this->assignment_model = new AssignmentModel();
26         $this->user = new User();
27         if ( $this->user->level <= 2) // permission denied
28             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
29     }
30
31
32
33
34 // -----
35
36
37
38
39     public function index()
40     {
41
42         $data = array(
43             'logs' => $this->logs_model->get_all_logs(),
44             'selected' => 'logs',
45             'user' => $this->user,
46             'all_assignments' => $this->assignment_model->all_assignments(),
47             'finish_time' => $this->user->selected_assignment['finish_time'],
48             'extra_time' => $this->user->selected_assignment['extra_time'],
49         );
50
51         return view('pages/admin/logs', $data);
52     }
53
54
55
56
57 // -----
58
59 }

```

Kode A.8: Moss.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4

```

```

4  * @file Moss.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Controllers;
8 use App\Controllers\BaseController;
9 use App\Models\AssignmentModel;
10 use App\Models\SettingsModel;
11 use App\Models\SubmitModel;
12 use App\Models\User;
13
14 class Moss extends BaseController
15 {
16
17     protected $validation;
18     protected $assignment_model;
19     protected $settings_model;
20     protected $user;
21     protected $submit_model;
22
23     public function __construct()
24     {
25         $this->validation = \Config\Services::validation();
26         $this->assignment_model = new AssignmentModel();
27         $this->settings_model = new SettingsModel();
28         $this->user = new User();
29     }
30
31
32 // -----
33
34
35     public function index($assignment_id = FALSE)
36     {
37         if ($assignment_id === FALSE)
38             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
39         $this->validation->setRule('detect', 'detect', 'required');
40         if ($this->validation->withRequest($this->request)->run())
41         {
42             if ($this->request->getPost('detect') !== 'detect')
43                 exit;
44             $this->_detect($assignment_id);
45         }
46         $data = array(
47             'all_assignments' => $this->assignment_model->all_assignments(),
48             'moss_userid' => $this->settings_model->get_setting('moss_userid'),
49             'moss_assignment' => $this->assignment_model->assignment_info($assignment_id),
50             'update_time' => $this->assignment_model->get_moss_time($assignment_id),
51             'selected' => 'assignments',
52             'user' => $this->user,
53             'finish_time' => $this->user->selected_assignment['finish_time'],
54             'extra_time' => $this->user->selected_assignment['extra_time'],
55         );
56
57         $data['moss_problems'] = array();
58         $assignments_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
59         for($i=1; $i<=$data['moss_assignment'][problems]; $i++){
60             $data['moss_problems'][$i] = NULL;
61             $path = $assignments_path."/assignment_{$assignment_id}/p{$i}/moss_link.txt";
62             if (file_exists($path))
63                 $data['moss_problems'][$i] = file_get_contents($path);
64         }
65
66         return view('pages/admin/moss', $data);
67     }
68
69
70 // -----
71
72
73     public function update($assignment_id = FALSE)
74     {
75         if ($assignment_id === FALSE)
76             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
77         $userid = $this->request->getPost('moss_userid');
78         $this->settings_model->set_setting('moss_userid', $userid);
79         $moss_original = trim(file_get_contents(rtrim($this->settings_model->get_setting('tester_path'), '/').'/moss_original'));
80         ;
81         $moss_path = rtrim($this->settings_model->get_setting('tester_path'), '/').'/moss';
82         file_put_contents($moss_path, str_replace('MOSS_USER_ID', $userid, $moss_original));
83         shell_exec("chmod +x {$moss_path}");
84         return $this->index($assignment_id);
85     }
86
87 // -----
88
89
90     private function _detect($assignment_id = FALSE)
91     {
92         if ($assignment_id === FALSE)
93             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
94         $this->submit_model = new SubmitModel();
95         $assignments_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
96         $tester_path = rtrim($this->settings_model->get_setting('tester_path'), '/');
97         shell_exec("chmod +x {$tester_path}/moss");
98         $items = $this->submit_model->get_final_submissions($assignment_id, $this->user->level, $this->user->username);
99         $groups = array();
100        foreach ($items as $item) {
101            if (!isset($groups[$item['problem']])))
102        }
103    }

```

```

102         $groups[$item['problem']] = array($item);
103     else
104         array_push($groups[$item['problem']], $item);
105     }
106     foreach ($groups as $problem_id => $group) {
107         $list = '';
108         $assignment_path = $assignments_path . "/assignment_{$assignment_id}";
109         foreach ($group as $item)
110             if ($item['file_type'] !== 'zip' && $item['file_type'] !== 'pdf')
111                 $list .= "{$problem_id}/{$item['username']}/{$item['file_name']}.".filetype_to_extension($item['file_type'])
112                 ]. ".";
113         shell_exec("list=$list; cd $assignment_path; $tester_path/moss_$list | grep http>p{$problem_id}/moss_link.txt");
114     }
115     $this->assignment_model->set_moss_time($assignment_id);
116 }
117
118 }
```

Kode A.9: Notifications.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4 * @file Notifications.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9
10 use App\Controllers\BaseController;
11 use App\Models\AssignmentModel;
12 use App\Models\NotificationsModel;
13 use App\Models\User;
14
15 class Notifications extends BaseController
16 {
17
18     private $notif_edit;
19     protected $session;
20     protected $validation;
21     protected $notifications_model;
22     protected $assignment_model;
23     protected $user;
24
25 // -----
26
27
28     public function __construct()
29     {
30         $this->session = session();
31         $this->validation = \Config\Services::validation();
32         $this->notifications_model = new NotificationsModel();
33         $this->assignment_model = new AssignmentModel();
34         $this->notif_edit = FALSE;
35         $this->user = new User();
36     }
37
38 // -----
39
40
41     public function index()
42     {
43         $data = array(
44             'all_assignments' => $this->assignment_model->all_assignments(),
45             'notifications' => $this->notifications_model->get_all_notifications(),
46             'user' => $this->user,
47             'selected' => 'notifications',
48             'finish_time' => $this->user->selected_assignment['finish_time'],
49             'extra_time' => $this->user->selected_assignment['extra_time'],
50         );
51
52         return view('pages/notifications', $data);
53
54     }
55
56 // -----
57
58
59     public function add()
60     {
61         if ( $this->user->level <=1){ // permission denied
62             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
63         }
64
65         $this->validation->setRule('title', 'title', 'trim');
66         $this->validation->setRule('text', 'text', 'trim'); /* todo: xss clean */
67
68         if ($this->request->is('post')){
69             if($this->validation->withRequest($this->request)->run()){
70                 if ($this->request->getPost('id') === NULL)
71                     $this->notifications_model->add_notification($this->request->getPost('title'), $this->request->getPost('text'))
72                         );
73                 else
74                     $this->notifications_model->update_notification($this->request->getPost('id'), $this->request->getPost('title')
75                         ), $this->request->getPost('text'));
76             }
77         }
78     }
79
80 }
```

```

76        return redirect('notifications');
77    }
78
79    $data = array(
80        'all_assignments' => $this->assignment_model->all_assignments(),
81        'notif_edit' => $this->notif_edit,
82        'user' => $this->user,
83        'selected' => 'notifications',
84        'finish_time' => $this->user->selected_assignment['finish_time'],
85        'extra_time' => $this->user->selected_assignment['extra_time'],
86    );
87
88    if ($this->notif_edit !== FALSE){
89        $data['title'] = 'Edit_Notification';
90    }
91
92    return view('pages/admin/add_notification',$data);
93}
94
95
96 // -----
97
98
99 public function edit($notif_id = FALSE)
100 {
101     if ($this->user->level <= 1) // permission denied
102         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
103     if ($notif_id === FALSE || ! is_numeric($notif_id))
104         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
105
106     $this->notif_edit = $this->notifications_model->get_notification($notif_id);
107
108     return $this->add();
109 }
110
111
112 // -----
113
114
115 public function delete()
116 {
117     if ( ! $this->request->isAJAX() )
118         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
119     if ($this->user->level <= 1) // permission denied
120         $json_result = array('done' => 0, 'message' => 'Permission_Denied');
121     elseif ($this->request->getPost('id') === NULL)
122         $json_result = array('done' => 0, 'message' => 'Input_Error');
123     else
124     {
125         $this->notifications_model->delete_notification($this->request->getPost('id'));
126         $json_result = array('done' => 1);
127     }
128
129     $this->response->setContentType('application/json', 'charset=utf-8');
130     echo json_encode($json_result);
131 }
132
133
134 // -----
135
136
137 public function check()
138 {
139     if ( ! $this->request->isAJAX() )
140         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
141     $time = $this->request->getPost('time');
142     if ($time === NULL)
143         exit('error');
144     if ($this->notifications_model->have_new_notification(strtotime("$time")))
145         exit('new_notification');
146     exit('no_new_notification');
147 }
148
149 }
150

```

Kode A.10: Problems.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4 * @file Problems.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9
10 use App\Controllers\BaseController;
11 use App\Models\AssignmentModel;
12 use App\Models\SettingsModel;
13 use App\Models\User;
14
15 class Problems extends BaseController
16 {
17
18     private $all_assignments;
19     protected $session;
20     protected $assignment_model;

```

```

21 protected $user;
22 protected $settings_model;
23 protected $validation;
24
25 // -----
26
27
28 public function __construct()
29 {
30     $this->session = session();
31     $this->assignment_model = new AssignmentModel();
32     $this->user = new User();
33     $this->settings_model = new SettingsModel();
34     $this->all_assignments = $this->assignment_model->all_assignments();
35     $this->validation = \Config\Services::validation();
36 }
37
38
39 // -----
40
41
42 /**
43 * Displays detail description of given problem
44 *
45 * @param int $assignment_id
46 * @param int $problem_id
47 */
48 public function index($assignment_id = NULL, $problem_id = 1)
49 {
50
51     // If no assignment is given, use selected assignment
52     if ($assignment_id === NULL)
53         $assignment_id = $this->user->selected_assignment['id'];
54     if ($assignment_id == 0)
55         throw new \CodeIgniter\Exceptions\PageNotFoundException('No_assignment_selected');
56
57     $assignment = $this->assignment_model->assignment_info($assignment_id);
58
59     $data = array(
60         'all_assignments' => $this->all_assignments,
61         'all_problems' => $this->assignment_model->all_problems($assignment_id),
62         'description_assignment' => $assignment,
63         'can_submit' => TRUE,
64         'selected' => 'problems',
65         'user' => $this->user,
66         'finish_time' => $this->user->selected_assignment['finish_time'],
67         'extra_time' => $this->user->selected_assignment['extra_time'],
68     );
69
70     if ( ! is_numeric($problem_id) || $problem_id < 1 || $problem_id > $data['description_assignment']['problems'])
71         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
72
73     $languages = explode(',',$data['all_problems'][$problem_id]['allowed_languages']);
74
75     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
76     $problem_dir = "$assignments_root/assignment_{$assignment_id}/p{$problem_id}";
77     $data['problem'] = array(
78         'id' => $problem_id,
79         'description' => '<p>Description not found</p>',
80         'allowed_languages' => $languages,
81         'has_pdf' => glob("$problem_dir/*.pdf") != FALSE
82     );
83
84     $path = "$problem_dir/desc.html";
85     if (file_exists($path))
86         $data['problem']['description'] = file_get_contents($path);
87
88     if ( $assignment['id'] == 0
89         OR ( $this->user->level == 0 && ! $assignment['open'] )
90         OR shj_now() < strtotime($assignment['start_time'])
91         OR shj_now() > strtotime($assignment['finish_time'])+$assignment['extra_time'] // deadline = finish_time + extra_time
92         OR ! $this->assignment_model->is_participant($assignment['participants'], $this->user->username)
93     )
94         $data['can_submit'] = FALSE;
95
96     return view('pages/problems', $data);
97 }
98
99 // -----
100
101
102
103 /**
104 * Edit problem description as html/markdown
105 *
106 * $type can be 'md', 'html', or 'plain'
107 *
108 * @param string $type
109 * @param int $assignment_id
110 * @param int $problem_id
111 */
112 public function edit($type = 'md', $assignment_id = NULL, $problem_id = 1)
113 {
114     if ($type !== 'html' && $type !== 'md' && $type !== 'plain')
115         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
116
117     if ($this->user->level <= 1)
118         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
119

```

```

120
121     switch($type)
122     {
123         case 'html':
124             $ext = 'html'; break;
125         case 'md':
126             $ext = 'md'; break;
127         case 'plain':
128             $ext = 'html'; break;
129     }
130
131     if ($assignment_id === NULL)
132         $assignment_id = $this->user->selected_assignment['id'];
133     if ($assignment_id == 0)
134         throw new \Exception('No_assignment_selected.');
135
136     $data = array(
137         'all_assignments' => $this->assignment_model->all_assignments(),
138         'description_assignment' => $this->assignment_model->assignment_info($assignment_id),
139         'selected' => 'problems',
140         'finish_time' => $this->user->selected_assignment['finish_time'],
141         'extra_time' => $this->user->selected_assignment['extra_time'],
142         'user' => $this->user,
143     );
144
145     if ( ! is_numeric($problem_id) || $problem_id < 1 || $problem_id > $data['description_assignment']['problems'])
146         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
147
148     $this->validation->setRule('text', 'text', 'required'); /* todo: xss clean */
149     if ($this->validation->withRequest($this->request)->run())
150     {
151         $this->assignment_model->save_problem_description($assignment_id, $problem_id, $this->request->getPost('text'), $ext);
152         return redirect()->to('problems/'.$assignment_id.'/'.$problem_id);
153     }
154
155     $data['problem'] = array(
156         'id' => $problem_id,
157         'description' => ''
158     );
159
160     $path = rtrim($this->settings_model->get_setting('assignments_root'), '/') . "/assignment_{$assignment_id}/p{$problem_id}/
161         desc.". $ext;
162     if (file_exists($path))
163         $data['problem']['description'] = file_get_contents($path);
164
165     return view('pages/admin/edit_problem_'.$type, $data);
166 }
167
168 }
169

```

Kode A.11: Profile.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Queue.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\QueueModel;
12 use App\Models\SettingsModel;
13 use App\Models\User;
14
15 class Queue extends BaseController
16 {
17     protected $queue_model;
18     protected $settings_model;
19     protected $assignment_model;
20     protected $submit_model;
21     protected $user;
22
23     public function __construct()
24     {
25         $this->queue_model = new QueueModel();
26         $this->settings_model = new SettingsModel();
27         $this->assignment_model = new AssignmentModel();
28         $this->user = new User();
29     }
30
31
32 // -----
33
34
35     public function index()
36     {
37
38         $data = array(
39             'all_assignments' => $this->assignment_model->all_assignments(),
40             'queue' => $this->queue_model->get_queue(),
41             'working' => $this->settings_model->get_setting('queue_is_working'),
42             'user' => $this->user,
43             'finish_time' => $this->user->selected_assignment['finish_time'],
44             'extra_time' => $this->user->selected_assignment['extra_time'],

```

```

45     'selected' => ''
46 );
47
48     return view('pages/admin/queue', $data);
49 }
50
51 // -----
52
53
54 public function pause()
55 {
56     if ( ! $this->request->isAJAX() ){
57         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
58     }else{
59         $this->settings_model->set_setting('queue_is_working','0');
60         echo 'success';
61     }
62 }
63
64
65 // -----
66
67
68 public function resume()
69 {
70     if ( ! $this->request->isAJAX() ){
71         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
72     }else{
73         process_the_queue();
74         echo 'success';
75     }
76 }
77
78 // -----
79
80
81
82 public function empty_queue()
83 {
84     if ( ! $this->request->isAJAX() ){
85         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
86     }else{
87         $this->queue_model->empty_queue();
88         echo 'success';
89     }
90 }
91 }
92 }
```

Kode A.12: Queue.php

```

1 <?php
2 /**
3 * Sharif Judge online judge
4 * @file Queue.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\QueueModel;
12 use App\Models\SettingsModel;
13 use App\Models\User;
14
15 class Queue extends BaseController
16 {
17     protected $queue_model;
18     protected $settings_model;
19     protected $assignment_model;
20     protected $submit_model;
21     protected $user;
22
23     public function __construct()
24     {
25         $this->queue_model = new QueueModel();
26         $this->settings_model = new SettingsModel();
27         $this->assignment_model = new AssignmentModel();
28         $this->user = new User();
29     }
30
31 // -----
32
33
34     public function index()
35     {
36
37         $data = array(
38             'all_assignments' => $this->assignment_model->all_assignments(),
39             'queue' => $this->queue_model->get_queue(),
40             'working' => $this->settings_model->get_setting('queue_is_working'),
41             'user' => $this->user,
42             'finish_time' => $this->user->selected_assignment['finish_time'],
43             'extra_time' => $this->user->selected_assignment['extra_time'],
44             'selected' => ''
45         );
46     }
47 }
```

```

48     return view('pages/admin/queue', $data);
49 }
50
51 // -----
52
53
54 public function pause()
55 {
56     if ( ! $this->request->isAJAX() ){
57         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
58     }else{
59         $this->settings_model->set_setting('queue_is_working', '0');
60         echo 'success';
61     }
62 }
63
64
65 // -----
66
67
68 public function resume()
69 {
70     if ( ! $this->request->isAJAX() ){
71         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
72     }else{
73         process_the_queue();
74         echo 'success';
75     }
76 }
77
78
79 // -----
80
81
82 public function empty_queue()
83 {
84     if ( ! $this->request->isAJAX() ){
85         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
86     }else{
87         $this->queue_model->empty_queue();
88         echo 'success';
89     }
90 }
91
92 }

```

Kode A.13: Queueprocess.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4 * @file Queueprocess.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\QueueModel;
12 use App\Models\SettingsModel;
13 use App\Models\SubmitModel;
14
15 class Queueprocess extends BaseController
16 {
17     protected $queue_model;
18     protected $submit_model;
19     protected $settings_model;
20     protected $assignment_model;
21     protected $db;
22     protected $config;
23
24     public function __construct()
25     {
26         // This controller should not be called from a browser
27         $this->queue_model = new QueueModel();
28         $this->settings_model = new SettingsModel();
29         $this->assignment_model = new AssignmentModel();
30         $this->submit_model = new SubmitModel();
31         $this->db = db_connect();
32     }
33
34
35
36 // -----
37
38 public function test()
39 {
40     $op1 = $this->settings_model->get_setting('enable_log');
41     echo ($op1);
42 }
43
44
45 /**
46 * This is the main function for processing the queue
47 * This function judges queue items one by one
48 */
49 public function run()
50

```

```

51 {
52
53     // Set correct base_url
54     // Because we are in cli mode, base_url is not available, and we get
55     // it from an environment variable that we have set in shj_helper.php
56     $this->config = config('App');
57     $this->config->baseURL = getenv('SHJ_BASE_URL');
58
59
60     $queue_item = $this->queue_model->get_first_item();
61     if ($queue_item === NULL) {
62         $this->settings_model->set_setting('queue_is_working', '0');
63         exit;
64     }
65     if ($this->settings_model->get_setting('queue_is_working')){
66         exit;
67     }
68
69     $this->settings_model->set_setting('queue_is_working', '1');
70
71     do { // loop over queue items
72
73         $submit_id = $queue_item['submit_id'];
74         $username = $queue_item['username'];
75         $assignment = $queue_item['assignment'];
76         $assignment_info = $this->assignment_model->assignment_info($assignment);
77         $problem = $this->assignment_model->problem_info($assignment, $queue_item['problem']);
78         $type = $queue_item['type']; // $type can be 'judge', 'rejudge', or 'exec'
79
80         $submission = $this->submit_model->get_submission($username, $assignment, $problem['id'], $submit_id);
81
82         $file_type = $submission['file_type'];
83         $file_extension = filetype_to_extension($file_type);
84         $raw_filename = $submission['file_name'];
85         $main_filename = $submission['main_file_name'];
86
87         $assignments_dir = rtrim($this->settings_model->get_setting('assignments_root'), '/');
88         $tester_path = rtrim($this->settings_model->get_setting('tester_path'), '/');
89         $problemdir = $assignments_dir."/assignment_$assignment/p".$problem['id'];
90         $userdir = "$problemdir/$username";
91         //$the_file = "$userdir/$raw_filename.$file_extension";
92
93         $op1 = $this->settings_model->get_setting('enable_log');
94         $op2 = $this->settings_model->get_setting('enable_easysandbox');
95         $op3 = 0;
96         if ($file_type === 'c')
97             $op3 = $this->settings_model->get_setting('enable_c_shield');
98         elseif ($file_type === 'cpp')
99             $op3 = $this->settings_model->get_setting('enable_cpp_shield');
100        $op4 = 0;
101        if ($file_type === 'py2')
102            $op4 = $this->settings_model->get_setting('enable_py2_shield');
103        elseif ($file_type === 'py3')
104            $op4 = $this->settings_model->get_setting('enable_py3_shield');
105        $op5 = $this->settings_model->get_setting('enable_java_policy');
106        $op6 = $assignment_info['javaexceptions'];
107
108        if($type === 'exec') {
109            $exec_only = TRUE;
110            $op7 = 1;
111        }
112        else {
113            $op7 = 0;
114        }
115
116        if ($file_type === 'c' OR $file_type === 'cpp')
117            $time_limit = $problem['c_time_limit']/1000;
118        elseif ($file_type === 'java')
119            $time_limit = $problem['java_time_limit']/1000;
120        elseif ($file_extension === 'py')
121            $time_limit = $problem['python_time_limit']/1000;
122        $time_limit = round($time_limit, 3);
123        $time_limit_int = floor($time_limit) + 1;
124
125        $memory_limit = $problem['memory_limit'];
126        $diff_cmd = $problem['diff_cmd'];
127        $diff_arg = $problem['diff_arg'];
128        $output_size_limit = $this->settings_model->get_setting('output_size_limit') * 1024;
129
130        $cmd = "cd_$tester_path;\n./tester.sh_$problemdir_".escapeshellarg($username).'_'.escapeshellarg($main_filename).'_'.
131           escapeshellarg($raw_filename)."_$file_type$time_limit$time_limit_int$memory_limit$output_size_limit_".
132           $diff_cmd_$diff_arg_$op1_$op2_$op3_$op4_$op5_$op6_$op7";
133
134        file_put_contents($userdir.'/log', $cmd);
135
136        /////////////////////////////////
137        // Running tester (judging the code) //
138        /////////////////////////////////
139        // putenv('LANG=en_US.UTF-8');
140        $output = trim(shell_exec($cmd));
141
142        // Deleting the jail folder, if still exists
143        shell_exec("cd_$tester_path;rm_rf_jail*");
144
145        // Saving judge result
146        if ( is_numeric($output) || $output === 'Compilation_Error' || $output === 'Syntax_Error' )
147        {
148            shell_exec("mv_$userdir/result.html_$userdir/result-{$submit_id}.html");
149            shell_exec("mv_$userdir/log_$userdir/log-{$submit_id}");

```

```

148     }
149
150     if (is_numeric($output)) {
151         $submission['pre_score'] = $output;
152         $submission['status'] = 'SCORE';
153     }
154     else {
155         $submission['pre_score'] = 0;
156         $submission['status'] = $output;
157     }
158
159     //reconnect to database incase we have run test for a long time.
160     $this->db->reconnect();
161     // Save the result
162     $this->queue_model->save_judge_result_in_db($submission, $type);
163
164     // Remove the judged item from queue
165     $this->queue_model->remove_item($username, $assignment, $problem['id'], $submit_id);
166
167     // Get next item from queue
168     $queue_item = $this->queue_model->get_first_item();
169
170     }while($queue_item !== NULL && $this->settings_model->get_setting('queue_is_working'));
171
172     $this->settings_model->set_setting('queue_is_working', '0');
173
174 }
175
176 }
```

Kode A.14: Rejudge.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Rejudge.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\QueueModel;
12 use App\Models\User;
13
14 class Rejudge extends BaseController
15 {
16
17     protected $problems;
18     protected $assignment_model;
19     protected $user;
20     protected $validation;
21     protected $queue_model;
22
23     public function __construct()
24     {
25         $this->assignment_model = new AssignmentModel();
26         $this->user = new User();
27         $this->validation = \Config\Services::validation();
28         $this->problems = $this->assignment_model->all_problems($this->user->selected_assignment['id']);
29     }
30
31
32     // -----
33
34
35     public function index()
36     {
37
38         $this->validation->setRule('problem_id', 'problem_id', 'required|integer');
39
40         $data = array(
41             'all_assignments' => $this->assignment_model->all_assignments(),
42             'problems' => $this->problems,
43             'msg' => array(),
44             'user' => $this->user,
45             'finish_time' => $this->user->selected_assignment['finish_time'],
46             'extra_time' => $this->user->selected_assignment['extra_time'],
47             'selected' => 'assignments'
48         );
49
50         if ($this->validation->withRequest($this->request)->run())
51         {
52             $problem_id = $this->request->getPost('problem_id');
53             $this->queue_model = new QueueModel();
54             $this->queue_model->rejudge($this->user->selected_assignment['id'], $problem_id);
55             process_the_queue();
56             $data['msg'] = array('Rejudge_in_progress');
57         }
58
59         return view('pages/admin/rejudge', $data);
60
61
62     // -----
63
64
65     public function rejudge_single()
66 }
```

```
67    {
68        if ( ! $this->request->isAJAX() )
69            throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
70
71        $this->validation->setRule('submit_id', 'submit_id', 'required|integer');
72        $this->validation->setRule('username', 'username', 'required|alpha_numeric');
73        $this->validation->setRule('assignment', 'assignment', 'required|integer');
74        $this->validation->setRule('problem', 'problem', 'required|integer');
75
76        if ($this->validation->withRequest($this->request)->run())
77        {
78            $this->queue_model = new QueueModel();
79            $this->queue_model->rejudge_single(
80                array(
81                    'submit_id' => $this->request->getPost('submit_id'),
82                    'username' => $this->request->getPost('username'),
83                    'assignment' => $this->request->getPost('assignment'),
84                    'problem' => $this->request->getPost('problem'),
85                )
86            );
87            process_the_queue();
88            $json_result = array('done' => 1);
89        }
90        else
91            $json_result = array('done' => 0, 'message' => 'Input_Error');
92
93        $this->response->setHeader('Content-Type:application/json;', 'charset=utf-8');
94        echo json_encode($json_result);
95    }
96
97 }
```

Kode A.15: Scoreboard.php

```
1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Scoreboard.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\ScoreboardModel;
12 use App\Models\User;
13
14 class Scoreboard extends BaseController
15 {
16     protected $scoreboard_model;
17     protected $assignment_model;
18     protected $user;
19
20     public function __construct()
21     {
22         $this->scoreboard_model = new ScoreboardModel();
23         $this->assignment_model = new AssignmentModel();
24         $this->user = new User();
25     }
26
27     public function index()
28     {
29         $data = array(
30             'all_assignments' => $this->assignment_model->all_assignments(),
31             'scoreboard' => $this->scoreboard_model->get_scoreboard($this->user->selected_assignment['id']),
32             'user' => $this->user,
33             'selected' => 'scoreboard',
34             'finish_time' => $this->user->selected_assignment['finish_time'],
35             'extra_time' => $this->user->selected_assignment['extra_time'],
36         );
37
38         return view('pages/scoreboard', $data);
39     }
40
41 }
42 }
```

Kode A.16: Server time.php

```
1 <?php
2 /**
3  * Sharif Judge online judge
4  * @file Server_time.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Controllers;
8
9 use App\Controllers\BaseController;
10
11 class Server_time extends BaseController
12 {
13     /**
14      * Prints server time, used for server time synchronization
15     */
16     public function index()
```

```

17|     {
18|         echo shj_now_str();
19|     }
20|

```

Kode A.17: Settings.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Settings.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9
10 use App\Controllers\BaseController;
11 use App\Models\AssignmentModel;
12 use App\Models\User;
13 use App\Models\SettingsModel;
14 use Config\Validation;
15 use DateTimeZone;
16
17 class Settings extends BaseController
18 {
19
20     private $form_status;
21     private $errors;
22     protected $session;
23     protected $user;
24     protected $settings_model;
25     protected $assignment_model;
26     protected $validation;
27     protected $validationError;
28
29
30 // -----
31
32
33     public function __construct()
34     {
35         $this->session = session();
36         $this->user = new User();
37         $this->settings_model = new SettingsModel();
38         $this->assignment_model = new AssignmentModel();
39         $this->validation = \Config\Services::validation();
40         $this->form_status = '';
41         $this->errors = array();
42         $this->validationError = null;
43     }
44
45
46 // -----
47
48
49     public function index()
50     {
51         $settings = $this->settings_model->get_all_settings();
52         $data = array_merge($settings,
53             array(
54                 'all_assignments' => $this->assignment_model->all_assignments(),
55                 'sandbox_built' => file_exists(rtrim($settings['tester_path'], '/') . '/easysandbox/EasySandbox.so'),
56                 'form_status' => $this->form_status,
57                 'errors' => $this->errors,
58                 'selected' => 'settings',
59                 'level' => $this->user->level,
60                 'finish_time' => $this->user->selected_assignment['finish_time'],
61                 'extra_time' => $this->user->selected_assignment['extra_time'],
62                 'user' => $this->user,
63                 'validationError' => $this->validation
64             )
65         );
66         ob_start();
67         $data ['defc'] = file_get_contents(rtrim($settings['tester_path'], '/') . '/shield/defc.h');
68         $data ['defcpp'] = file_get_contents(rtrim($settings['tester_path'], '/') . '/shield/defcpp.h');
69         $data ['shield_py2'] = file_get_contents(rtrim($settings['tester_path'], '/') . '/shield/shield_py2.py');
70         $data ['shield_py3'] = file_get_contents(rtrim($settings['tester_path'], '/') . '/shield/shield_py3.py');
71         ob_end_clean();
72
73         return view('pages/admin/settings', $data);
74     }
75
76
77 // -----
78
79
80     public function update()
81     {
82         $this->validation->setRule('timezone', 'timezone', 'required');
83         $this->validation->setRule('file_size_limit', 'File_size_limit', 'integer|greater_than_equal_to[0]');
84         $this->validation->setRule('output_size_limit', 'Output_size_limit', 'integer|greater_than_equal_to[0]');
85         $this->validation->setRule('rpp_all', 'results_per_page_(all_submissions)', 'integer|greater_than_equal_to[0]');
86         $this->validation->setRule('rpp_final', 'results_per_page_(final_submissions)', 'integer|greater_than_equal_to[0]');
87         $this->validation->setRule('mail_from', 'email', 'valid_email');
88         if($this->validation->withRequest($this->request)->run()){
89             ob_start();
90             $this->form_status = 'ok';
91             $tester_path = rtrim($this->settings_model->get_setting('tester_path'), '/');

```

```

92 $defc_path = $tester_path.'/shield/defc.h';
93 $defcpp_path = $tester_path.'/shield/defcpp.h';
94 $shpy2_path = $tester_path.'/shield/shield_py2.py';
95 $shpy3_path = $tester_path.'/shield/shield_py3.py';
96 if ($this->request->getVar('def_c') !== file_get_contents($defc_path))
97     if (file_exists($defc_path) && file_put_contents($defc_path,$this->request->getVar('def_c')) === FALSE)
98         array_push($this->errors, 'File_defc.h_is_not_writable._Edit_it_manually.');
99     if ($this->request->getVar('def_cpp') !== file_get_contents($defcpp_path))
100        if (file_exists($defcpp_path) && file_put_contents($defcpp_path,$this->request->getVar('def_cpp')) === FALSE)
101            array_push($this->errors, 'File_defcpp.h_is_not_writable._Edit_it_manually.');
102    if ($this->request->getVar('shield_py2') !== file_get_contents($shpy2_path))
103        if (file_exists($shpy2_path) && file_put_contents($shpy2_path,$this->request->getVar('shield_py2')) === FALSE)
104            array_push($this->errors, 'File_shield_py2.py_is_not_writable._Edit_it_manually.');
105    if ($this->request->getVar('shield_py3') !== file_get_contents($shpy3_path))
106        if (file_exists($shpy3_path) && file_put_contents($shpy3_path,$this->request->getVar('shield_py3')) === FALSE)
107            array_push($this->errors, 'File_shield_py3.py_is_not_writable._Edit_it_manually.');
108 ob_end_clean();
109 $timezone = $this->request->getVar('timezone');
110 // if timezone is invalid, set it to 'Asia/Tehran' :
111 if (!in_array($timezone, DateTimeZone::listIdentifiers()) )
112     $timezone='Asia/Tehran';
113
114 $this->settings_model->set_settings(
115     array(
116         'timezone' => $timezone,
117         'tester_path' => $this->request->getVar('tester_path'),
118         'assignments_root' => $this->request->getVar('assignments_root'),
119         'file_size_limit' => $this->request->getVar('file_size_limit'),
120         'output_size_limit' => $this->request->getVar('output_size_limit'),
121         'default_late_rule' => $this->request->getVar('default_late_rule'),
122         'enable_easysandbox' => $this->request->getVar('enable_easysandbox') === NULL?0:1,
123         'enable_c_shield' => $this->request->getVar('enable_c_shield') === NULL?0:1,
124         'enable_cpp_shield' => $this->request->getVar('enable_cpp_shield') === NULL?0:1,
125         'enable_py2_shield' => $this->request->getVar('enable_py2_shield') === NULL?0:1,
126         'enable_py3_shield' => $this->request->getVar('enable_py3_shield') === NULL?0:1,
127         'enable_java_policy' => $this->request->getVar('enable_java_policy') === NULL?0:1,
128         'enable_log' => $this->request->getVar('enable_log') === NULL?0:1,
129         'enable_registration' => $this->request->getVar('enable_registration') === NULL?0:1,
130         'registration_code' => $this->request->getVar('registration_code'),
131         'mail_from' => $this->request->getVar('mail_from'),
132         'mail_from_name' => $this->request->getVar('mail_from_name'),
133         'reset_password_mail' => $this->request->getVar('reset_password_mail'),
134         'add_user_mail' => $this->request->getVar('add_user_mail'),
135         'results_per_page_all' => $this->request->getVar('rpp_all'),
136         'results_per_page_final' => $this->request->getVar('rpp_final'),
137         'week_start' => $this->request->getVar('week_start'),
138         'lock_student_display_name' => $this->request->getVar('lock_student_display_name') === NULL?0:1,
139     )
140 );
141 }
142 else{
143     $this->form_status = 'error';
144 }
145 return $this->index();
146 }
147 }

```

Kode A.18: Submissions.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Submissions.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\SettingsModel;
12 use App\Models\SubmitModel;
13 use App\Models\User;
14 use App\Models\UserModel;
15 use App\Libraries\Shj_pagination;
16 use App\Models\ScoreboardModel;
17 use PhpOffice\PhpSpreadsheet\Spreadsheet;
18 use PhpOffice\PhpSpreadsheet\IOFactory;
19 use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
20 use PhpOffice\PhpSpreadsheet\Style\Fill;
21 use PhpOffice\PhpSpreadsheet\Style\Border;
22 use PhpOffice\PhpSpreadsheet\Style\Alignment;
23
24 class Submissions extends BaseController
25 {
26
27     private $problems;
28
29     private $filter_user;
30     private $filter_problem;
31     private $page_number;
32
33     protected $assignment_model;
34     protected $submit_model;
35     protected $user;
36     protected $validation;
37     protected $settings_model;
38     protected $user_model;
39     protected $scoreboard_model;

```

```

40 protected $uri;
41
42 // -----
43
44 public function __construct()
45 {
46     $this->user = new User();
47     $this->assignment_model = new AssignmentModel();
48     $this->submit_model = new SubmitModel();
49     $this->validation = \Config\Services::validation();
50     $this->settings_model = new SettingsModel();
51     $this->user_model = new UserModel();
52     $this->problems = $this->assignment_model->all_problems($this->user->selected_assignment['id']);
53     $this->uri = service('uri');
54     $temp_inp = $this->uri->getSegments();
55     $this->filter_user = $this->filter_problem = NULL;
56     $this->page_number = 1;
57     $input=[];
58
59     foreach($temp_inp as $key => $value){
60         if($key != (count($temp_inp)-1)){
61             $input[$value] = $temp_inp[$key+1];
62         }else{
63             break;
64         }
65     }
66     if (array_key_exists('user', $input) && $input['user']){
67         if ($this->user->level > 0) // students are not able to filter submissions by user
68             $this->filter_user = ctype_alnum($input['user'])? $input['user']:NULL;
69     if (array_key_exists('problem', $input) && $input['problem']){
70         $this->filter_problem = is_numeric($input['problem'])? $input['problem']:NULL;
71     if (array_key_exists('page', $input) && $input['page']){
72         $this->page_number = is_numeric($input['page'])? $input['page']:1;
73     }
74 }
75
76
77
78 // -----
79
80
81 /**
82 * Uses PHPExcel library to generate excel file of submissions
83 */
84 private function _download_excel($view)
85 {
86     if ( ! in_array($view, array('all', 'final')))
87         exit;
88
89     $now = shj_now_str(); // current time
90
91     // Load PHPExcel library
92     $phpspreadsheet = new Spreadsheet();
93
94     // Set document properties
95     $phpspreadsheet->getProperties()->setCreator('SharIF_Judge')
96         ->setLastModifiedBy('SharIF_Judge')
97         ->setTitle('SharIF_Judge_Users')
98         ->setSubject('SharIF_Judge_Users')
99         ->setDescription('List_of_ShariF_Judge_users_(.'.$now.')');
100
101    // Name of the file sent to browser
102    $output_filename = 'judge_'.$view.'_submissions';
103
104    // Set active sheet
105    $phpspreadsheet->setActiveSheetIndex(0);
106    $sheet = $phpspreadsheet->getActiveSheet();
107
108    // Add current assignment, time, username filter, and problem filter to document
109    $sheet->fromArray(array('Assignment:', $this->user->selected_assignment['name']), null, 'A1', true);
110    $sheet->fromArray(array('Time: ', $now), null, 'A2', true);
111    $sheet->fromArray(array('Username_Filter:', $this->filter_user? $this->filter_user:'No_filter'), null, 'A3', true);
112    $sheet->fromArray(array('Problem_Filter:', $this->filter_problem? $this->filter_problem:'No_filter'), null, 'A4', true);
113
114    // Prepare header
115    if ($this->user->level === 0)
116        $header=array('Final','Problem','Submit_Time','Score','Delay_(HH:MM)','Coefficient','Final_Score','Language','Status')
117        ;
118    else{
119        $header=array('Final','Submit_ID','Username','Name','Problem','Submit_Time','Score','Delay_(HH:MM)','Coefficient',
120        'Final_Score','Language','Status');
121        if ($view === 'final'){
122            array_unshift($header, "#2");
123            array_unshift($header, "#1");
124        }
125    }
126
127    // Add header to document
128    $sheet->fromArray($header, null, 'A6', true);
129    $highest_column = $sheet->getHighestColumn();
130
131    // Set custom style for header
132    $sheet->getStyle('A6:'.$highest_column.'6')->applyFromArray(
133        array(
134            'fill' => array(
135                'fillType' => Fill::FILL_SOLID,
136                'color' => array('rgb' => '173C45')
137            ),
138            'font' => array(
139                'bold' => true
140            )
141        )
142    );

```

```

137         'font' => array(
138             'bold' => true,
139             'color' => array('rgb' => 'FFFFFF'),
140             // 'size' => 14
141         )
142     );
143 );
144
145 // Prepare data (in $rows array)
146 if ($view === 'final')
147     $items = $this->submit_model->get_final_submissions($this->user->selected_assignment['id'], $this->user->level, $this->user->username, NULL, $this->filter_user, $this->filter_problem);
148 else
149     $items = $this->submit_model->get_all_submissions($this->user->selected_assignment['id'], $this->user->level, $this->user->username, NULL, $this->filter_user, $this->filter_problem);
150
151 $names = $this->user_model->get_names();
152
153 $finish = strtotime($this->user->selected_assignment['finish_time']);
154 $i=0; $j=0; $un='';
155 $rows = array();
156 foreach ($items as $item){
157     $i++;
158     if ($item['username'] != $un)
159         $j++;
160     $un = $item['username'];
161
162     $pi = $this->problems[$item['problem']];
163
164     $pre_score = ceil($item['pre_score']*$pi['score']/10000);
165
166     $checked='';
167     if ($item['is_final'])
168         $checked='*';
169
170     $delay = strtotime($item['time'])-$finish;
171     if ($item['coefficient'] === 'error')
172         $final_score = 0;
173     else
174         $final_score = ceil($pre_score*$item['coefficient']/100);
175
176     if ($this->user->level === 0)
177         $row = array(
178             $checked,
179             $item['problem'].'.'.$pi['name'],
180             $item['time'],
181             $pre_score,
182             ($delay==0?'No_Delay':time_hhmm($delay)),
183             $item['coefficient'],
184             $final_score,
185             filetype_to_language($item['file_type']),
186             $item['status'],
187         );
188     else {
189         $row = array(
190             $checked,
191             $item['submit_id'],
192             $item['username'],
193             $names[$item['username']],
194             $item['problem'].'.'.$pi['name'],
195             $item['time'],
196             $pre_score,
197             ($delay==0?'No_Delay':time_hhmm($delay)),
198             $item['coefficient'],
199             $final_score,
200             filetype_to_language($item['file_type']),
201             $item['status'],
202         );
203         if ($view === 'final'){
204             array_unshift($row,$j);
205             array_unshift($row,$i);
206         }
207     }
208     array_push($rows, $row);
209 }
210
211 // Add rows to document
212 $sheet->fromArray($rows, null, 'A7', true);
213 // Add alternative colors to rows
214 for ($i=7; $i<count($rows)+7; $i++){
215     $sheet->getStyle('A'.$i.':'.$highest_column.$i)->applyFromArray(
216         array(
217             'fill' => array(
218                 'fillType' => Fill::FILL_SOLID,
219                 'color' => array('rgb' => (($i%2)?'F0F0F0':'FAFAFA'))
220             )
221         )
222     );
223 }
224
225 // Set text align to center
226 $sheet->getStyle( $sheet->calculateWorksheetDimension() )
227     ->getAlignment()
228     ->setHorizontal(Alignment::HORIZONTAL_CENTER);
229
230 // Making columns autosize
231 for ($i=2;$i<count($header);$i++)
232     $sheet->getColumnDimension(chr(65+$i))->setAutoSize(true);

```

```

234 // Set Border
235 $sheet->getStyle('A7:'.$highest_column.$sheet->getHighestRow())->applyFromArray(
236     array(
237         'borders' => array(
238             'outline' => array(
239                 'borderStyle' => Border::BORDER_THIN,
240                 'color' => array('rgb' => '444444'),
241             ),
242         ),
243     )
244 );
245 );
246
247 // Send the file to browser
248
249 $ext = 'xlsx';
250 if ( ! class_exists('ZipArchive') ) // If class ZipArchive does not exist, export to excel5 instead of excel 2007
251     $ext = 'xls';
252
253 header('Content-Type:_application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
254 header('Content-Disposition:_attachment;filename="'.$output_filename.'.'.$ext.'"');
255 header('Cache-Control:_max-age=0');
256 $objWriter = IOFactory::createWriter($phpspreadsheet, ucfirst($ext));
257 $objWriter->save('php://output');
258 }
259
260
261
262
263 // -----
264
265
266
267
268 public function final_excel()
269 {
270     $this->download_excel('final');
271 }
272
273
274
275 public function all_excel()
276 {
277     $this->download_excel('all');
278 }
279
280
281
282
283 // -----
284
285
286
287
288 public function the_final()
289 {
290
291     if ( ! is_numeric($this->page_number) )
292         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
293
294     if ($this->page_number<1)
295         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
296
297     $config = array(
298         'base_url' => site_url('submissions/final'.($this->filter_user?'/user/'.$this->filter_user:'')).($this->filter_problem?
299             '/problem/'.$this->filter_problem:''),
300         'cur_page' => $this->page_number,
301         'total_rows' => $this->submit_model->count_final_submissions($this->user->selected_assignment['id'], $this->user->
302             level, $this->user->username, $this->filter_user, $this->filter_problem),
303         'per_page' => $this->settings_model->get_setting('results_per_page_final'),
304         'num_links' => 5,
305         'full_ul_class' => 'shj_pagination',
306         'cur_li_class' => 'current_page'
307     );
308
309     if ($config['per_page'] == 0)
310         $config['per_page'] = $config['total_rows'];
311     $shj_pagination = new Shj_pagination($config);
312
313     $submissions = $this->submit_model->get_final_submissions($this->user->selected_assignment['id'], $this->user->level,
314             $this->user->username, $this->page_number, $this->filter_user, $this->filter_problem);
315
316     $names = $this->user_model->get_names();
317
318     foreach ($submissions as &$item)
319     {
320         $item['name'] = $names[$item['username']];
321         $item['fullmark'] = ($item['pre_score'] == 10000);
322         $item['pre_score'] = ceil($item['pre_score']*{$this->problems[$item['problem']]}}['score']/10000);
323         $item['delay'] = strtotime($item['time'])-strtotime($this->user->selected_assignment['finish_time']);
324         $item['language'] = filetype_to_language($item['file_type']);
325         if ($item['coefficient'] === 'error')
326             $item['final_score'] = 0;
327         else
328             $item['final_score'] = ceil($item['pre_score']*$item['coefficient']/100);
329     }
330
331     $data = array(

```

```

330     'view' => 'final',
331     'all_assignments' => $this->assignment_model->all_assignments(),
332     'problems' => $this->problems,
333     'submissions' => $submissions,
334     'excel_link' => site_url('submissions/final_excel'.($this->filter_user?'/user/'.$this->filter_user:'').($this->
335         filter_problem?'/problem/'.$this->filter_problem:'')),
336     'filter_user' => $this->filter_user,
337     'filter_problem' => $this->filter_problem,
338     'pagination' => $shj_pagination->create_links(),
339     'page_number' => $this->page_number,
340     'per_page' => $config['per_page'],
341     'selected' => "final_submissions",
342     'user' => $this->user,
343     'finish_time' => $this->user->selected_assignment['finish_time'],
344     'extra_time' => $this->user->selected_assignment['extra_time'],
345   );
346 
347   return view('pages/submissions', $data);
348 
349 
350 
351 
352 // -----
353 
354 
355 
356 
357 public function all()
358 {
359 
360   if ( ! is_numeric($this->page_number))
361     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
362 
363   if ($this->page_number < 1)
364     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
365 
366   $config = array(
367     'base_url' => site_url('submissions/all'.($this->filter_user?'/user/'.$this->filter_user:'').($this->filter_problem?'/
368         problem/'.$this->filter_problem:'')),
369     'cur_page' => $this->page_number,
370     'total_rows' => $this->submit_model->count_all_submissions($this->user->selected_assignment['id'], $this->user->level,
371       $this->user->username, $this->filter_user, $this->filter_problem),
372     'per_page' => $this->settings_model->get_setting('results_per_page_all'),
373     'num_links' => 5,
374     'full_ul_class' => 'shj_pagination',
375     'cur_li_class' => 'current_page'
376   );
377   if ($config['per_page']==0)
378     $config['per_page'] = $config['total_rows'];
379 
380   $shj_pagination = new Shj_pagination($config);
381   $submissions = $this->submit_model->get_all_submissions($this->user->selected_assignment['id'], $this->user->level, $this-
382     ->user->username, $this->page_number, $this->filter_user, $this->filter_problem);
383   $names = $this->user_model->get_names();
384 
385   foreach ($submissions as &$item)
386   {
387     $item['name'] = $names[$item['username']];
388     $item['fullmark'] = ($item['pre_score'] == 10000);
389     $item['pre_score'] = ceil($item['pre_score']*($this->problems[$item['problem']]['score']/10000));
390     $item['delay'] = strtotime($item['time'])-strtotime($this->user->selected_assignment['finish_time']);
391     $item['language'] = filetype_to_language($item['file_type']);
392     if ($item['coefficient'] === 'error')
393       $item['final_score'] = 0;
394     else
395       $item['final_score'] = ceil($item['pre_score']*$item['coefficient']/100);
396   }
397 
398   $data = array(
399     'view' => 'all',
400     'all_assignments' => $this->assignment_model->all_assignments(),
401     'problems' => $this->problems,
402     'submissions' => $submissions,
403     'excel_link' => site_url('submissions/all_excel'.($this->filter_user?'/user/'.$this->filter_user:'').($this->
404         filter_problem?'/problem/'.$this->filter_problem:'')),
405     'filter_user' => $this->filter_user,
406     'filter_problem' => $this->filter_problem,
407     'pagination' => $shj_pagination->create_links(),
408     'selected' => "all_submissions",
409     'user' => $this->user,
410     'finish_time' => $this->user->selected_assignment['finish_time'],
411     'extra_time' => $this->user->selected_assignment['extra_time'],
412   );
413 
414   return view('pages/submissions', $data);
415 
416 // -----
417 
418 
419 
420 
421 /**
422 * Used by ajax request (for selecting final submission)
423 */

```

```

424 public function select()
425 {
426     if ( ! $this->request->isAJAX() )
427         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
428
429     // Students cannot change their final submission after finish_time + extra_time
430     if ($this->user->level === 0)
431         if ( shj_now() > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time'])
432     {
433         $json_result = array(
434             'done' => 0,
435             'message' => 'This_assignment_is_finished._You_cannot_change_your_final_submissions.'
436         );
437         $this->response->setHeader('Content-Type: application/json;', 'charset=utf-8');
438         echo json_encode($json_result);
439         return;
440     }
441
442     $this->validation->setRule('submit_id', 'Submit_ID', 'integer|greater_than[0]');
443     $this->validation->setRule('problem', 'problem', 'integer|greater_than[0]');
444     $this->validation->setRule('username', 'Username', 'required|min_length[3]|max_length[20]|alpha_numeric');
445
446     if ($this->validation->withRequest($this->request)->run())
447     {
448         $username = $this->request->getPost('username');
449         if ($this->user->level === 0)
450             $username = $this->user->username;
451
452         $res = $this->submit_model->set_final_submission(
453             $username,
454             $this->user->selected_assignment['id'],
455             $this->request->getPost('problem'),
456             $this->request->getPost('submit_id')
457         );
458
459         if ($res) {
460             // each time a user changes final submission, we should update scoreboard of that assignment
461             $this->scoreboard_model = new ScoreboardModel();
462             $this->scoreboard_model->update_scoreboard($this->user->selected_assignment['id']);
463             $json_result = array('done' => 1);
464         }
465         else
466             $json_result = array('done' => 0, 'message' => 'Selecting_Final_Submission_Failed');
467     }
468     else
469         $json_result = array('done' => 0, 'message' => 'Input_Error');
470
471     $this->response->setHeader('Content-Type: application/json;', 'charset=utf-8');
472     echo json_encode($json_result);
473 }
474
475 /**
476 * For "view code" or "view result" or "view log"
477 */
478
479 public function view_code()
480 {
481     if ( ! $this->request->isAJAX() )
482         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
483     $this->validation->setRule('type','type','check_type');
484     $this->validation->setRule('username','username','required|min_length[3]|max_length[20]|alpha_numeric');
485     $this->validation->setRule('assignment','assignment','integer|greater_than[0]');
486     $this->validation->setRule('problem','problem','integer|greater_than[0]');
487     $this->validation->setRule('submit_id','submit_id','integer|greater_than[0]');
488
489     if($this->validation->withRequest($this->request)->run())
490     {
491         $submission = $this->submit_model->get_submission(
492             $this->request->getPost('username'),
493             $this->request->getPost('assignment'),
494             $this->request->getPost('problem'),
495             $this->request->getPost('submit_id')
496         );
497         if ($submission === FALSE)
498             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
499
500         $type = $this->request->getPost('type'); // $type is 'code', 'result', or 'log'
501
502         if ($this->user->level === 0 && $type === 'log')
503             throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
504
505         if ($this->user->level === 0 && $this->user->username != $submission['username'])
506             exit('Don\'t_try_to_see_submitted_codes;');
507
508         if ($type === 'result')
509             $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
510             "/assignment_{$submission['assignment']}/p{$submission['problem']}/{$submission['username']}/result-{$
511                 $submission['submit_id']}.html";
512         elseif ($type === 'code')
513             $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
514             "/assignment_{$submission['assignment']}/p{$submission['problem']}/{$submission['username']}/{$submission['
515                 file_name']}.{$filetype_to_extension($submission['file_type'])}";
516         elseif ($type === 'log')
517             $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
518             "/assignment_{$submission['assignment']}/p{$submission['problem']}/{$submission['username']}/log-{$submission
519                 ['submit_id']}";
520         else
521             $file_path = '/nowhere'; // This line is never reached!

```

```

519     $result = array(
520         'file_name' => $submission['main_file_name'].'.'.filetype_to_extension($submission['file_type']),
521         'text' => file_exists($file_path)?file_get_contents($file_path):'File Not Found'
522     );
523
524     if ($type === 'code') {
525         $result['lang'] = $submission['file_type'];
526         if ($result['lang'] == 'py2' || $result['lang'] == 'py3')
527             $result['lang'] = 'python';
528     }
529
530     $this->response->setContentType('application/json')->setBody(json_encode($result));
531     $this->response->send();
532
533 }
534 else
535     exit('Are you trying to see other users\' codes?');
536
537 }
538
539
540
541
542 // -----
543
544
545
546
547 public function download_file()
548 {
549     $username = $this->uri->getSegment(3);
550     $assignment = $this->uri->getSegment(4);
551     $problem = $this->uri->getSegment(5);
552     $submit_id = $this->uri->getSegment(6);
553
554     $submission = $this->submit_model->get_submission(
555         $username,
556         $assignment,
557         $problem,
558         $submit_id
559     );
560     if ($submission === FALSE)
561         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
562
563     if ($this->user->level === 0 && $this->user->username != $submission['username'])
564         exit('Don\'t try to see submitted codes!');
565
566     $file_path = rtrim($this->settings_model->get_setting('assignments_root'), '/');
567     "/$assignment->{$submission['assignment']}/{$submission['problem']}/{$submission['username']}/{$submission['file_name']}." .
568         filetype_to_extension($submission['file_type']);
569
570     return $this->response->download(
571         "{$submission['file_name']}.".filetype_to_extension($submission['file_type']),
572         file_get_contents($file_path)
573     );
574 }
575
576
577 }

```

Kode A.19: Submit.php

```

1 <?php
2 /**
3 * Sharif Judge online judge
4 * @file Submit.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7
8 namespace App\Controllers;
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\QueueModel;
12 use App\Models\SettingsModel;
13 use App\Models\SubmitModel;
14 use App\Models\User;
15
16 class Submit extends BaseController
17 {
18
19     private $data; //data sent to view
20     private $assignment_root;
21     private $problems;
22     private $problem;//submitted problem id
23     private $filetype; //type of submitted file
24     private $ext; //uploaded file extension
25     private $file_name; //uploaded file name without extension
26     private $coefficient;
27
28 // -----
29     protected $queue_model;
30     protected $settings_model;
31     protected $assignment_model;
32     protected $submit_model;
33     protected $user;
34     protected $validation;
35

```

```

36|     public function __construct()
37|     {
38|         $this->queue_model = new QueueModel();
39|         $this->settings_model = new SettingsModel();
40|         $this->assignment_model = new AssignmentModel();
41|         $this->user = new User();
42|         $this->validation = \Config\Services::validation();
43|         $this->assignment_root = $this->settings_model->get_setting('assignments_root');
44|         $this->problems = $this->assignment_model->all_problems($this->user->selected_assignment['id']);
45|
46|         helper('shj_helper');
47|         $extra_time = $this->user->selected_assignment['extra_time'];
48|         $delay = shj_now() - strtotime($this->user->selected_assignment['finish_time']);
49|         ob_start();
50|         if ( isset($this->user->selected_assignment['late_rule']) ? eval($this->user->selected_assignment['late_rule']) === FALSE
51|             : '' )
52|             $coefficient = "error";
53|         if (!isset($coefficient))
54|             $coefficient = "error";
55|         ob_end_clean();
56|         $this->coefficient = $coefficient;
57|
58|     }
59|
60|
61|     public function _language_to_type($language)
62|     {
63|         $language = strtolower ($language);
64|         switch ($language) {
65|             case 'c': return 'c';
66|             case 'c++': return 'cpp';
67|             case 'python_2': return 'py2';
68|             case 'python_3': return 'py3';
69|             case 'java': return 'java';
70|             case 'zip': return 'zip';
71|             case 'pdf': return 'pdf';
72|             case 'txt': return 'txt';
73|             default: return FALSE;
74|         }
75|     }
76|
77|
78|     public function _language_to_ext($language)
79|     {
80|         $language = strtolower ($language);
81|         switch ($language) {
82|             case 'c': return 'c';
83|             case 'c++': return 'cpp';
84|             case 'python_2': return 'py';
85|             case 'python_3': return 'py';
86|             case 'java': return 'java';
87|             case 'zip': return 'zip';
88|             case 'pdf': return 'pdf';
89|             case 'txt': return 'txt';
90|             default: return FALSE;
91|         }
92|     }
93|
94|
95|     public function _match($type, $extension)
96|     {
97|         switch ($type) {
98|             case 'c': return ($extension==='c'?TRUE:FALSE);
99|             case 'cpp': return ($extension==='cpp'?TRUE:FALSE);
100|             case 'py2': return ($extension==='py'?TRUE:FALSE);
101|             case 'py3': return ($extension==='py'?TRUE:FALSE);
102|             case 'java': return ($extension==='java'?TRUE:FALSE);
103|             case 'zip': return ($extension==='zip'?TRUE:FALSE);
104|             case 'pdf': return ($extension==='pdf'?TRUE:FALSE);
105|             case 'txt': return ($extension==='txt'?TRUE:FALSE);
106|         }
107|     }
108|
109|
110|     public function index()
111|     {
112|         $this->validation->setRule('problem', 'problem', 'required|integer|greater_than[0]', array('greater_than' => 'Select_a_%s.'));
113|         $this->validation->setRule('language', 'language', 'required|check_language', array('_check_language' => 'Select_a_valid_%s.'));
114|         if($this->request->is('post')){
115|             if ($this->validation->withRequest($this->request)->run())
116|             {
117|                 if ($this->upload())
118|                     return redirect()->to('submissions/all');
119|                 else
120|                     throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Error_Uploading_File:_'.$this->validation->listErrors());
121|             }
122|         }
123|     }
124|
125|
126|
127|
128|
129|
130|

```

```

131     }
132 }
133
134 $this->data = array(
135     'all_assignments' => $this->assignment_model->all_assignments(),
136     'problems' => $this->problems,
137     'in_queue' => FALSE,
138     'coefficient' => $this->coefficient,
139     'upload_state' => '',
140     'problems_js' => '',
141     'error' => '',
142     'selected' => 'submit',
143     'user' => $this->user,
144     'finish_time' => $this->user->selected_assignment['finish_time'],
145     'extra_time' => $this->user->selected_assignment['extra_time'],
146     'validationError' => $this->validation
147 );
148 foreach ($this->problems as $problem)
149 {
150     $languages = explode(',', $problem['allowed_languages']);
151     $items='';
152     foreach ($languages as $language)
153     {
154         $items = $items."\"".trim($language)."\",";
155     }
156     $items = substr($items,0,strlen($items)-1);
157     $this->data['problems_js'] .= "shj.p[{$problem['id']}]=[$items];_";
158 }
159 if ($this->user->selected_assignment['id'] == 0)
160     $this->data['error']= 'Please_select_an_assignment_first.';
161 elseif ($this->user->level == 0 && ! $this->user->selected_assignment['open'])
162     // if assignment is closed, non-student users (admin, instructors) still can submit
163     $this->data['error']= 'Selected_assignment_is_closed.';
164 elseif (shj_now() < strtotime($this->user->selected_assignment['start_time']))
165     $this->data['error']= 'Selected_assignment_has_not_started.';
166 elseif (shj_now() > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time']) // deadline = finish_time + extra_time
167     $this->data['error']= 'Selected_assignment_has_finished.';
168 elseif ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->username))
169     $this->data['error']= 'You_are_not_registered_for_submitting.';
170 else
171     $this->data['error']= 'none';
172
173 return view('pages/submit', $this->data);
174 }
175
176 // -----
177
178 /**
179 * Saves submitted code and adds it to queue for judging
180 */
181 private function _upload()
182 {
183     $now = shj_now();
184     foreach($this->problems as $item)
185     {
186         if ($item['id'] == $this->request->getPost('problem'))
187         {
188             $this->problem = $item;
189             break;
190         }
191     }
192     $filetype = $this->language_to_type(strtolower(trim($this->request->getPost('language'))));
193     $this->ext = substr(strrchr($_FILES['userfile']['name'],'.'),1); // uploaded file extension
194     $this->file_name = basename($_FILES['userfile']['name'], ".{$this->ext}"); // uploaded file name without extension
195     if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $this->problem['id']) )
196         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('You_have_already_submitted_for_this_problem._Your_last_submission_is_still_in_queue.');
197     if ($this->user->level==0 && !$this->user->selected_assignment['open'])
198         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_been_closed.');
199     if ( $now < strtotime($this->user->selected_assignment['start_time']) )
200         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_not_started.');
201     if ( $now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['extra_time'] )
202         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Selected_assignment_has_finished.');
203     if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->username) )
204         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('You_are_not_registered_for_submitting.');
205     $filetypes = explode(",",$this->problem['allowed_languages']);
206     foreach ($filetypes as $ filetype)
207     {
208         $filetype = $this->language_to_type(strtolower(trim($filetype)));
209     }
210     if ($_FILES['userfile']['error'] == 4)
211         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('No_file_chosen.');
212     if ( ! in_array($this->filetype, $filetypes) )
213         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('This_file_type_is_not_allowed_for_this_problem.');
214     if ( ! $this->_match($this->filetype, $this->ext) )
215         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('This_file_type_does_not_match_your_selected_language.');
216     if ( ! preg_match('/^([a-zA-Z0-9\-\_])+$/ ', $this->file_name) )
217         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('Invalid_characters_in_file_name.');
218
219     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'.$this->user->selected_assignment['id'].'/'.$this->problem['id'].'/.'.$this->user->username;
220     if ( !file_exists($user_dir) )
221         mkdir($user_dir, 0700);
222
223     $config['max_size'] = $this->settings_model->get_setting('file_size_limit');

```

```

224     $config['file_name'] = $this->file_name.".".($this->user->selected_assignment['total_submits']+1).".". $this->ext;
225     $config['max_file_name'] = 20;
226     $this->validation->setRule('userfile', 'userfile', 'max_size[userfile,'.$config['max_size'].']', ['max_size' => 'The file',
227         'you_are_attempting_to_upload_is_larger_than_the_permitted_size.']);
228     if ($this->validation->withRequest($this->request)->run())
229     {
230         $results = $this->request->getFile('userfile');
231         $results->move($user_dir,$config['file_name']);
232         $result = preg_replace('/\A.[^.]+'."/","",$results->getName());
233         $this->submit_model = new SubmitModel();
234
235         $submit_info = array(
236             'submit_id' => $this->assignment_model->increase_total_submits($this->user->selected_assignment['id']),
237             'username' => $this->user->username,
238             'assignment' => $this->user->selected_assignment['id'],
239             'problem' => $this->problem['id'],
240             'file_name' => $result,
241             'main_file_name' => $this->file_name,
242             'file_type' => $this->filetype,
243             'coefficient' => $this->coefficient,
244             'pre_score' => 0,
245             'time' => shj_now_str(),
246         );
247         if ($this->problem['is_upload_only'] == 0)
248         {
249             $this->queue_model->add_to_queue($submit_info);
250             process_the_queue();
251         }
252         else
253         {
254             $this->submit_model->add_upload_only($submit_info);
255         }
256
257         return TRUE;
258     }
259     return FALSE;
260 }
261
262 // -----
263 /**
264 * Load code from editor file
265 */
266 public function load($problem_id){
267     $user_dir = rtrim($this->assignment_root, '/')."/assignment_". $this->user->selected_assignment['id'].'/p' . $problem_id.'/';
268     $this->user->username;
269     $file_path = $user_dir.'/.EDITOR_FILE_NAME.'.EDITOR_FILE_EXT;
270     $input_path = $user_dir.'/.EDITOR_IN_NAME.'.EDITOR_FILE_EXT;
271     $output_path = $user_dir.'/.EDITOR_OUT_NAME.'.EDITOR_FILE_EXT;
272
273     if(!write_file($input_path, 'y')){}
274     if(!write_file($output_path, 'y')){}
275     if (!file_exists($file_path)){
276         $response = json_encode(array('content'=>'', 'message'=>'No_saved_file'));
277     }
278     else{
279         $file_content = file_get_contents($file_path);
280         if ($file_content === FALSE){
281             $response = json_encode(array('content'=>'', 'message'=>'Unable_to_load'));
282         }
283         else{
284             addslashes($file_content);
285             $response = json_encode(array('content'=>$file_content, 'message'=>'Loaded'));
286         }
287     }
288     echo $response;
289 }
290
291 // -----
292 /**
293 * Save code to editor file and submit/execute if needed
294 */
295 public function save($type = FALSE){
296     $data = $_POST['code_editor'];
297     $problem_id = $_POST['problem_id'];
298     $language = $_POST['language'];
299
300     $user_dir = rtrim($this->assignment_root, '/')."/assignment_". $this->user->selected_assignment['id'].'/p' . $problem_id.'/';
301     $this->user->username;
302     if (!file_exists($user_dir)){
303         mkdir($user_dir, 0700);
304     }
305     $file_path = $user_dir.'/.EDITOR_FILE_NAME.'.EDITOR_FILE_EXT;
306     $input_path = $user_dir.'/.EDITOR_IN_NAME.'.EDITOR_FILE_EXT;
307
308     if (!write_file($file_path, $data)){
309         $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_save'));
310         echo $response;
311     }
312     else{
313         $response = json_encode(array('status'=>TRUE, 'message'=>'Saved'));
314         if($type === FALSE){
315             echo $response;
316         }
317     }
318 }
319

```

```

320
321     $now = shj_now();
322     if ( $this->queue_model->in_queue($this->user->username,$this->user->selected_assignment['id'], $problem_id)){
323         $response = json_encode(array('status'=>FALSE, 'message'=>'You have already submitted for this problem. Your
324                                         last submission is still in queue.'));
325         echo $response;
326     }
327     else if ($this->user->level==0 && !$this->user->selected_assignment['open']){
328         $response = json_encode(array('status'=>FALSE, 'message'=>'Selected assignment has been closed.'));
329         echo $response;
330     }
331     else if ($now < strtotime($this->user->selected_assignment['start_time'])){
332         $response = json_encode(array('status'=>FALSE, 'message'=>'Selected assignment has not started.'));
333         echo $response;
334     }
335     else if ($now > strtotime($this->user->selected_assignment['finish_time'])+$this->user->selected_assignment['
336                                         extra_time']){
337         $response = json_encode(array('status'=>FALSE, 'message'=>'Selected assignment has finished.'));
338         echo $response;
339     }
340     else if ( ! $this->assignment_model->is_participant($this->user->selected_assignment['participants'],$this->user->
341                                         username)){
342         $response = json_encode(array('status'=>FALSE, 'message'=>'You are not registered for submitting.'));
343         echo $response;
344     }
345     else{
346         if($type === 'submit'){
347             $this->submit($data, $problem_id, $language, $user_dir);
348         }
349         else if($type === 'execute'){
350             $editor_input = $_POST['editor_input'];
351             if (!write_file($input_path, $editor_input)){
352                 $response = json_encode(array('status'=>FALSE, 'message'=>'Unable to write input file'));
353                 echo $response;
354             }
355         }
356     }
357 }
358
359
360 }
361
362
363 // -----
364 /**
365 * Add code to queue for judging
366 */
367
368 private function _submit($data, $problem_id, $language, $user_dir){
369     $file_type = $this->language_to_type(strtolower(trim($language)));
370     $file_ext = $this->language_to_ext(strtolower(trim($language)));
371     $file_name = EDITOR_FILE_NAME;
372     $file_fname = $file_name.'_'.$this->user->selected_assignment['total_submits']+1;
373     $file_path = $user_dir.'/'.$file_fname.$file_ext;
374
375     foreach($this->problems as $item)
376     {
377         if ($item['id'] == $problem_id)
378         {
379             $this->problem = $item;
380             break;
381         }
382
383         if (!write_file($file_path, $data)){
384             $response = json_encode(array('status'=>FALSE, 'message'=>'Unable to submit'));
385         }
386         else{
387             $this->submit_model = new SubmitModel();
388
389             $submit_info = array(
390                 'submit_id' => $this->assignment_model->increase_total_submits($this->user->selected_assignment['id']),
391                 'username' => $this->user->username,
392                 'assignment' => $this->user->selected_assignment['id'],
393                 'problem' => $problem_id,
394                 'file_name' => $file_fname,
395                 'main_file_name' => $file_name,
396                 'file_type' => $file_type,
397                 'coefficient' => $this->coefficient,
398                 'pre_score' => 0,
399                 'time' => shj_now_str(),
400             );
401             if ($this->problem['is_upload_only'] == 0)
402             {
403                 $this->queue_model->add_to_queue($submit_info);
404                 process_the_queue();
405             }
406             else
407             {
408                 $this->submit_model->add_upload_only($submit_info);
409             }
410
411             $response = json_encode(array('status'=>TRUE, 'message'=>"Submitted"));
412         }
413         echo $response;
414     }
415 }
```

```

416 // -----
417 /**
418 * Add code to queue for execution only
419 */
420 private function _execute($data, $problem_id, $language, $user_dir){
421     $file_type = $this->language_to_type(strtolower(trim($language)));
422     $file_ext = $this->language_to_ext(strtolower(trim($language)));
423     $file_name = EDITOR_FILE_NAME;
424     $file_fname = $file_name.'_'.$EDITOR_SUBMIT_ID;
425     $file_path = $user_dir.'/'.$file_fname.'.'.$file_ext;
426     $output_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
427
428     if (!write_file($file_path, $data)){
429         $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_execute', 'debug'=>$file_path));
430     } else{
431         $submit_info = array(
432             'submit_id' => EDITOR_SUBMIT_ID,
433             'username' => $this->user->username,
434             'assignment' => $this->user->selected_assignment['id'],
435             'problem' => $problem_id,
436             'file_name' => $file_fname,
437             'main_file_name' => $file_name,
438             'file_type' => $file_type,
439             'coefficient' => $this->coefficient,
440             'pre_score' => 0,
441             'time' => shj_now_str(),
442         );
443
444         if($this->queue_model->add_to_queue_exec($submit_info)){
445             if (!write_file($output_path, 'Queueing...')){
446                 $response = json_encode(array('status'=>FALSE, 'message'=>'Unable_to_write_output_file'));
447             } else{
448                 process_the_queue();
449                 $response = json_encode(array('status'=>TRUE, 'message'=>'Executing'));
450             }
451         } else{
452             $response = json_encode(array('status'=>FALSE, 'message'=>'Still_in_queue'));
453         }
454     }
455     echo $response;
456 }
457
458 // -----
459 /**
460 * Load output file as execution result
461 */
462 public function get_output($problem_id){
463     $user_dir = rtrim($this->assignment_root, '/').'/assignment_'. $this->user->selected_assignment['id'].'/p'.'/'.$problem_id.'/';
464     $user_dir .= $this->user->username;
465     $file_path = $user_dir.'/'.$EDITOR_OUT_NAME.'.'.$EDITOR_FILE_EXT;
466
467     if (!file_exists($file_path)){
468         $response = json_encode(array('status'=>FALSE, 'content'=>''));
469     } else{
470
471         $file_content = file_get_contents($file_path);
472         if ($file_content === FALSE){
473             $response = json_encode(array('status'=>FALSE, 'content'=>''));
474         } else{
475             $complete_status = strpos($file_content, 'Total_Execution_Time');
476             if($complete_status === FALSE){
477                 $response = json_encode(array('status'=>FALSE, 'content'=>$file_content));
478             } else{
479                 $response = json_encode(array('status'=>TRUE, 'content'=>$file_content));
480             }
481         }
482     }
483     echo $response;
484 }
485
486 }
487
488
489
490
491
492 }
493

```

Kode A.20: Users.php

```

1 <?php
2 /**
3 * SharIF Judge online judge
4 * @file Users.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7 namespace App\Controllers;
8
9 use App\Controllers\BaseController;
10 use App\Models\AssignmentModel;
11 use App\Models\User;
12 use App\Models\UserModel;
13 use PhpOffice\PhpSpreadsheet\Spreadsheet;
14 use PhpOffice\PhpSpreadsheet\IOFactory;
15 use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
16 use PhpOffice\PhpSpreadsheet\Style\Fill;

```

```

17| use PhpOffice\PhpSpreadsheet\Style\Border;
18| use PhpOffice\PhpSpreadsheet\Style\Alignment;
19|
20 class Users extends BaseController
21 {
22
23     protected $session;
24     protected $user;
25     protected $user_model;
26     protected $assignment_model;
27     protected $validation;
28
29     public function __construct()
30     {
31         $this->session = session();
32         $this->user = new User();
33         $this->user_model = new UserModel();
34         $this->assignment_model = new AssignmentModel();
35         $this->validation = \Config\Services::validation();
36     }
37     // -----
38
39     public function index()
40     {
41
42         $data = array(
43             'all_assignments' => $this->assignment_model->all_assignments(),
44             'users' => $this->user_model->get_all_users(),
45             'selected' => 'users',
46             'level' => $this->user->level,
47             'finish_time' => $this->user->selected_assignment['finish_time'],
48             'extra_time' => $this->user->selected_assignment['extra_time'],
49             'user' => $this->user,
50         );
51
52         return view('pages/admin/users', $data);
53     }
54
55
56
57
58     // -----
59
60
61
62
63     public function add()
64     {
65         $data = array(
66             'all_assignments' => $this->assignment_model->all_assignments(),
67             'selected' => 'users',
68             'level' => $this->user->level,
69             'finish_time' => $this->user->selected_assignment['finish_time'],
70             'extra_time' => $this->user->selected_assignment['extra_time'],
71             'user' => $this->user,
72         );
73         $this->validation->setRule('new_users', 'New_Users', 'required');
74         if ($this->validation->withRequest($this->request)->run())
75         {
76             if ( !$this->request->isAJAX() ){
77                 exit;
78             }else{
79                 list($ok, $error) = $this->user_model->add_users(
80                     $this->request->getPost('new_users'),
81                     $this->request->getPost('send_mail'),
82                     $this->request->getPost('delay')
83                 );
84                 return view('pages/admin/add_user_result', array('ok' => $ok, 'error' => $error));
85             }
86         }
87         else
88         {
89             return view('pages/admin/add_user', $data);
90         }
91     }
92
93
94
95
96     // -----
97
98
99
100
101 /**
102 * Controller for deleting a user
103 * Called by ajax request
104 */
105 public function delete()
106 {
107     if ( ! $this->request->isAJAX() )
108         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
109     $user_id = $this->request->getPost('user_id');
110     if ( ! is_numeric($user_id) )
111         $json_result = array('done' => 0, 'message' => 'Input_Error');
112     elseif ($this->user_model->delete_user($user_id))
113         $json_result = array('done' => 1);
114     else
115         $json_result = array('done' => 0, 'message' => 'Deleting_User_Failed');

```

```

116     $this->response->setHeader('Content-Type', 'application/json; charset=utf-8');
117     echo json_encode($json_result);
118 }
119
120
121
122
123 // -----
124
125
126
127
128
129 /**
130 * Controller for deleting a user's submissions
131 * Called by ajax request
132 */
133 public function delete_submissions()
134 {
135     if ( ! $this->request->isAJAX() )
136         throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound('');
137     $user_id = $this->request->getPost('user_id');
138     if ( ! is_numeric($user_id) )
139         $json_result = array('done' => 0, 'message' => 'Input_Error');
140     elseif ($this->user_model->delete_submissions($user_id))
141         $json_result = array('done' => 1);
142     else
143         $json_result = array('done' => 0, 'message' => 'Deleting_Submissions_Failed');
144
145     $this->response->setHeader('Content-Type', 'application/json; charset=utf-8');
146     echo json_encode($json_result);
147 }
148
149
150
151
152 // -----
153
154
155
156
157 /**
158 * Uses PHPExcel library to generate excel file of users list
159 */
160 public function list_excel()
161 {
162
163     $now = shj_now_str(); // current time
164
165     // Load PHPExcel library
166     $phpspreadsheet = new Spreadsheet();
167
168     // Set document properties
169     $phpspreadsheet->getProperties()->setCreator('SharIF_Judge')
170         ->setLastModifiedBy('SharIF_Judge')
171         ->setTitle('SharIF_Judge_Users')
172         ->setSubject('SharIF_Judge_Users')
173         ->setDescription('List_of_ShariF_Judge_Users('. $now .')');
174
175     // Name of the file sent to browser
176     $output_filename = 'sharifjudge_users';
177
178     // Set active sheet
179     $phpspreadsheet->setActiveSheetIndex(0);
180     $sheet = $phpspreadsheet->getActiveSheet();
181
182     // Add current time to document
183     $sheet->fromArray(array('Time:' , $now), null, 'A1', true);
184
185     // Add header to document
186     $header = array('#', 'User_ID', 'Username', 'Display_Name', 'Email', 'Role', 'First_Login', 'Last_Login');
187     $sheet->fromArray($header, null, 'A3', true);
188     $highest_column = $sheet->getHighestColumn();
189
190     // Set custom style for header
191     $sheet->getStyle('A3:'.$highest_column.'3')->applyFromArray(
192         array(
193             'fill' => array(
194                 'fillType' => Fill::FILL_SOLID,
195                 'color' => array('rgb' => '173C45')
196             ),
197             'font' => array(
198                 'bold' => true,
199                 'color' => array('rgb' => 'FFFFFF'),
200                 // 'size' => 14
201             )
202         )
203     );
204
205     // Prepare user data (in $rows array)
206     $users = $this->user_model->get_all_users();
207     $i=0;
208     $rows = array();
209     foreach ($users as $user){
210         array_push($rows, array(
211             ++$i,
212             $user['id'],
213             $user['username'],
214             $user['display_name'],
215         )
216     );
217
218     // Add footer
219     $sheet->fromArray(array('Total Rows' => count($rows)), null, 'A1', true);
220
221     // Set file name
222     $output_filename = 'sharifjudge_users.xlsx';
223
224     // Output file
225     $writer = $phpspreadsheet->createWriter();
226     $writer->save($output_filename);
227
228     // Return file
229     return $output_filename;
230 }

```

```

215     $user['email'],
216     $user['role'],
217     $user['first_login_time']==NULL?'Never':$user['first_login_time'],
218     $user['last_login_time']==NULL?'Never':$user['last_login_time']
219   ));
220 }
221
222 // Add rows to document and set a background color of #7BD1BE
223 $sheet->fromArray($rows, null, 'A4', true);
224 // Add alternative colors to rows
225 for ($i=4; $i<count($rows)+4; $i++){
226   $sheet->getStyle('A'.$i.':'.$highest_column.$i)->applyFromArray(
227     array(
228       'fill' => array(
229         'fillType' => Fill::FILL_SOLID,
230         'color' => array('rgb' => (($i%2)?'F0F0F0':'FAFAFA'))
231       )
232     );
233 }
234
235
236 // Set text align to center
237 $sheet->getStyle( $sheet->calculateWorksheetDimension() )
238   ->getAlignment()
239   ->setHorizontal(Alignment::HORIZONTAL_CENTER);
240
241 // Making columns autosize
242 for ($i=2;$i<count($header);$i++)
243   $sheet->getColumnDimension(chr(65+$i))->setAutoSize(true);
244
245 // Set Border
246 $sheet->getStyle('A4:'.$highest_column.$sheet->getHighestRow())->applyFromArray(
247   array(
248     'borders' => array(
249       'outline' => array(
250         'borderStyle' => Border::BORDER_THIN,
251         'color' => array('rgb' => '444444'),
252       ),
253     ),
254   );
255 }
256
257 // Send the file to browser
258
259 // If class ZipArchive exists, export to excel2007, otherwise export to excel5
260 if ( class_exists('ZipArchive') )
261   $ext = 'xlsx';
262 else
263   $ext = 'xls';
264
265 header('Content-Type:_application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
266 header('Content-Disposition:_attachment;filename="'.$output_filename.'.'.$ext.'"');
267 header('Cache-Control:_max-age=0');
268 $objWriter = IOFactory::createWriter($phpspreadsheet, ucfirst($ext));
269 $objWriter->save('php://output');
270
271
272 }
273 }
```

LAMPIRAN B

KODE PROGRAM *MODELS*

Kode B.1: AssignmentModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Assignment_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7 namespace App\Models;
8
9 use App\Libraries\Parsedown;
10 use CodeIgniter\Model;
11 use App\Models\SettingsModel;
12 use App\Models\ScoreboardModel;
13
14 class AssignmentModel extends Model
15 {
16     protected $settings_model;
17     protected $scoreboard_model;
18     protected $request;
19     protected $parsedown;
20
21     public function __construct()
22     {
23         parent::__construct();
24         $this->settings_model = new SettingsModel();
25         $this->request = \Config\Services::request();
26     }
27
28     /**
29      * Add New Assignment to DB / Edit Existing Assignment
30      *
31      * @param $id
32      * @param bool $edit
33      * @return bool
34      */
35     public function add_assignment($id, $edit = FALSE)
36     {
37         $this->scoreboard_model = new ScoreboardModel();
38         // Start Database Transaction
39         $this->db->transStart();
40
41         $extra_items = explode('*', $this->request->getPost('extra_time'));
42         $extra_time = 1;
43         foreach($extra_items as $extra_item)
44         {
45             $extra_time *= $extra_item;
46         }
47
48         $archived_assignment = $archived_assignment = $this->request->getPost('archived_assignment')!=NULL ? 1 : 0;
49
50         $assignment = array(
51             'id' => $id,
52             'name' => $this->request->getPost('assignment_name'),
53             'problems' => $this->request->getPost('number_of_problems'),
54             'total_submits' => 0,
55             'open' => ($this->request->getPost('open')==NULL?0:1),
56             'scoreboard' => ($this->request->getPost('scoreboard')==NULL?0:1),
57             'javaexceptions' => ($this->request->getPost('javaexceptions')==NULL?0:1),
58             'description' => '', /* todo */
59             'start_time' => date('Y-m-d H:i:s', strtotime($this->request->getPost('start_time'))),
60             'finish_time' => date('Y-m-d H:i:s', strtotime($this->request->getPost('finish_time'))),
61             'extra_time' => $extra_time*60,
62             'late_rule' => $this->request->getPost('late_rule'),
63             'participants' => $this->request->getPost('participants'),
64             'archived_assignment' => $archived_assignment
65         );
66         if($edit)
67         {
68             $before = $this->db->table('assignments')->getWhere(['id'=>$id])->getRowArray();
69             unset($assignment['total_submits']);
70             $this->db->table('assignments')->where('id', $id)->update($assignment);
71             // each time we edit an assignment, we should update coefficient of all submissions of that assignment
72             if ($assignment['extra_time']==$before['extra_time'] OR $assignment['start_time']==$before['start_time'] OR
73                 $assignment['finish_time']==$before['finish_time'] OR $assignment['late_rule']==$before['late_rule'])
74                 $this->_update_coefficients($id, $assignment['extra_time'], $assignment['finish_time'], $assignment['late_rule']);
75         }
76     }
77 }
```

```

75|     else
76|         $this->db->table('assignments')->insert($assignment);
77|
78|     /* **** Adding problems to "problems" table **** */
79|
80|     //First remove all previous problems
81|     $this->db->table('problems')->delete(['assignment'=>$id]);
82|
83|     //Now add new problems:
84|     $names = $this->request->getPost('name');
85|     $scores = $this->request->getPost('score');
86|     $c_tl = $this->request->getPost('c_time_limit');
87|     $py_tl = $this->request->getPost('python_time_limit');
88|     $java_tl = $this->request->getPost('java_time_limit');
89|     $ml = $this->request->getPost('memory_limit');
90|     $ft = $this->request->getPost('languages');
91|     $dc = $this->request->getPost('diff_cmd');
92|     $da = $this->request->getPost('diff_arg');
93|     $uo = $this->request->getPost('is_upload_only');
94|     if ($uo === NULL)
95|         $uo = array();
96|     for ($i=1; $i<=$this->request->getPost('number_of_problems'); $i++)
97|     {
98|         $items = explode(',', $ft[$i-1]);
99|         $ft[$i-1] = '';
100|        foreach ($items as $item){
101|            $item = trim($item);
102|            $item2 = strtolower($item);
103|            $item = ucfirst($item2);
104|            if ($item2 === 'python2')
105|                $item = 'Python_2';
106|            elseif ($item2 === 'python3')
107|                $item = 'Python_3';
108|            elseif ($item2 === 'pdf')
109|                $item = 'PDF';
110|            $item2 = strtolower($item);
111|            if (! in_array($item2, array('c', 'c++', 'python_2', 'python_3', 'java', 'zip', 'pdf', 'txt')))
112|                continue;
113|            // If the problem is not Upload-Only, its language should be one of {C,C++,Python 2, Python 3,Java}
114|            if (! in_array($i, $uo) && ! in_array($item2, array('c', 'c++', 'python_2', 'python_3', 'java')) )
115|                continue;
116|            $ft[$i-1] .= $item . ",";
117|        }
118|        $ft[$i-1] = substr($ft[$i-1],0,strlen($ft[$i-1])-1); // remove last ','
119|        $problem = array(
120|            'assignment' => $id,
121|            'id' => $i,
122|            'name' => $names[$i-1],
123|            'score' => $scores[$i-1],
124|            'is_upload_only' => in_array($i,$uo)?1:0,
125|            'c_time_limit' => $c_tl[$i-1],
126|            'python_time_limit' => $py_tl[$i-1],
127|            'java_time_limit' => $java_tl[$i-1],
128|            'memory_limit' => $ml[$i-1],
129|            'allowed_languages' => $ft[$i-1],
130|            'diff_cmd' => $dc[$i-1],
131|            'diff_arg' => $da[$i-1],
132|        );
133|        $this->db->table('problems')->insert($problem);
134|    }
135|
136|    if ($edit)
137|    {
138|        // We must update scoreboard of the assignment
139|        $this->scoreboard_model->update_scoreboard($id);
140|    }
141|
142|    // Complete Database Transaction
143|    $this->db->transComplete();
144|
145|    return $this->db->transStatus();
146|
147|
148|
149|
150| // -----
151|
152|
153|
154| /**
155|  * Delete An Assignment
156|  *
157|  * @param $assignment_id
158|  */
159| public function delete_assignment($assignment_id)
160| {
161|     $this->db->transStart();
162|
163|     // Phase 1: Delete this assignment and its submissions from database
164|     $this->db->table('assignments')->delete(['id'=>$assignment_id]);
165|     $this->db->table('problems')->delete(['assignment'=>$assignment_id]);
166|     $this->db->table('submissions')->delete(['assignment'=>$assignment_id]);
167|
168|     $this->db->transComplete();
169|
170|     if ($this->db->transStatus())
171|     {
172|         // Phase 2: Delete assignment's folder (all test cases and submitted codes)
173|         $cmd = 'rm -rf '.rtrim($this->settings_model->get_setting('assignments_root'), '/').'/assignment_'.assignment_id;

```

```

174     shell_exec($cmd);
175 }
176
177
178
179
180 // -----
181
182
183
184 /**
185 * All Assignments
186 *
187 * Returns a list of all assignments and their information
188 *
189 * @return mixed
190 */
191 public function all_assignments()
192 {
193     $result = $this->db->table('assignments')->orderBy('id')->get()->getResultArray();
194     $assignments = [];
195     foreach ($result as $item)
196     {
197         $assignments[$item['id']] = $item;
198     }
199     return $assignments;
200 }
201
202
203
204 // -----
205
206
207
208 /**
209 * New Assignment ID
210 *
211 * Finds the smallest integer that can be used as id for a new assignment
212 *
213 * @return int
214 */
215 public function new_assignment_id()
216 {
217     $max = ($this->db->table('assignments')->selectMax('id', 'max_id')->get()->getRow()->max_id) + 1;
218
219     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
220     while (file_exists($assignments_root.'/'.$assignment_.$max)){
221         $max++;
222     }
223
224     return $max;
225 }
226
227
228
229 // -----
230
231
232
233 /**
234 * All Problems of an Assignment
235 *
236 * Returns an array containing all problems of given assignment
237 *
238 * @param $assignment_id
239 * @return mixed
240 */
241 public function all_problems($assignment_id)
242 {
243     $result = $this->db->table('problems')->orderBy('id')->getWhere(['assignment'=>$assignment_id])->getResultArray();
244     $problems = array();
245     foreach ($result as $row)
246     {
247         $problems[$row['id']] = $row;
248     }
249
250
251
252 // -----
253
254
255
256 /**
257 * Problem Info
258 *
259 * Returns database row for given problem (from given assignment)
260 *
261 * @param $assignment_id
262 * @param $problem_id
263 * @return mixed
264 */
265 public function problem_info($assignment_id, $problem_id)
266 {
267     return $this->db->table('problems')->getWhere(['assignment'=>$assignment_id, 'id'=>$problem_id])->getRowArray();
268 }
269
270
271
272 // -----

```

```

273
274
275
276 /**
277 * Assignment Info
278 *
279 * Returns database row for given assignment
280 *
281 * @param $assignment_id
282 * @return array
283 */
284 public function assignment_info($assignment_id)
285 {
286     $query = $this->db->table('assignments')->getWhere(['id'=>$assignment_id]);
287     if ($query->getNumRows() != 1){
288         return array(
289             'id' => 0,
290             'name' => 'Not_Selected',
291             'finish_time' => 0,
292             'extra_time' => 0,
293             'problems' => 0
294         );
295     }else{
296         return $query->getRowArray();
297     }
298 }
299
300
301
302 // -----
303
304
305 /**
306 * Is Participant
307 *
308 * Returns TRUE if $username if one of the $participants
309 * Examples for participants: "ALL" or "user1, user2,user3"
310 *
311 * @param $participants
312 * @param $username
313 * @return bool
314 */
315 public function is_participant($participants, $username)
316 {
317     $participants = explode(',', $participants);
318     foreach ($participants as &$participant){
319         $participant = trim($participant);
320     }
321     if(in_array('ALL', $participants))
322         return TRUE;
323     if(in_array($username, $participants))
324         return TRUE;
325     return FALSE;
326 }
327
328
329
330
331 // -----
332
333
334
335 /**
336 * Increase Total Submits
337 *
338 * Increases number of total submits for given assignment by one
339 *
340 * @param $assignment_id
341 * @return mixed
342 */
343 public function increase_total_submits($assignment_id)
344 {
345     // Get total submits
346     $total = $this->db->table('assignments')->select('total_submits')->getWhere(['id'=>$assignment_id])->getRow()->
347     total_submits;
348     // Save total+1 in DB
349     $this->db->table('assignments')->where('id', $assignment_id)->update(['total_submits'=>($total+1)]);
350
351     // Return new total
352     return ($total+1);
353 }
354
355
356 // -----
357
358
359
360 /**
361 * Set Moss Time
362 *
363 * Updates "Moss Update Time" for given assignment
364 *
365 * @param $assignment_id
366 */
367 public function set_moss_time($assignment_id)
368 {
369     $now = shj_now_str();
370     $this->db->table('assignments')->where('id', $assignment_id)->update(['moss_update'=>$now]);

```

```

371     }
372
373
374
375 // -----
376
377
378 /**
379 * Get Moss Time
380 *
381 * Returns "Moss Update Time" for given assignment
382 *
383 * @param $assignment_id
384 * @return string
385 */
386
387 public function get_moss_time($assignment_id)
388 {
389     $query = $this->db->table('assignments')->select('moss_update')->getWhere(['id'=>$assignment_id]);
390     if($query->getNumRows() != 1) return 'Never';
391     return $query->getRow()->moss_update;
392 }
393
394
395 // -----
396
397
398 /**
399 * Save Problem Description
400 *
401 * Saves (Adds/Updates) problem description (html or markdown)
402 *
403 * @param $assignment_id
404 * @param $problem_id
405 * @param $text
406 * @param $type
407 */
408
409 public function save_problem_description($assignment_id, $problem_id, $text, $type)
410 {
411     $assignments_root = rtrim($this->settings_model->get_setting('assignments_root'), '/');
412
413     if ($type === 'html')
414     {
415         // Remove the markdown code
416         if(file_exists("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md")){
417             unlink("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md");
418         }
419         // Save the html code
420         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.html", $text);
421     }
422     elseif ($type === 'md')
423     {
424         // We parse markdown using Parsedown library
425         $this->parsedown = new Parsedown;
426         // Save the markdown code
427         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.md", $text);
428         // Convert markdown to html and save the html
429         file_put_contents("$assignments_root/assignment_{$assignment_id}/p{$problem_id}/desc.html", $this->parsedown->parse(
430             $text));
431     }
432 }
433
434
435 // -----
436
437
438 /**
439 * Update Coefficients
440 *
441 * Each time we edit an assignment (Update start time, finish time, extra time, or
442 * coefficients rule), we should update coefficients of all submissions of that assignment
443 *
444 * This function is called from add_assignment($id, TRUE)
445 *
446 * @param $assignment_id
447 * @param $extra_time
448 * @param $finish_time
449 * @param $new_late_rule
450 */
451
452 private function _update_coefficients($assignment_id, $extra_time, $finish_time, $new_late_rule)
453 {
454     $submissions = $this->db->table('submissions')->getWhere(['assignment'=>$assignment_id])->getResultArray();
455
456     $finish_time = strtotime($finish_time);
457
458     foreach ($submissions as $i => $item) {
459         $delay = strtotime($item['time'])-$finish_time;
460         ob_start();
461         if ( eval($new_late_rule) === FALSE )
462             $coefficient = "error";
463         if (!isset($coefficient))
464             $coefficient = "error";
465         ob_end_clean();
466         $submissions[$i]['coefficient'] = $coefficient;
467     }
468 // For better performance, we update each 1000 rows in one SQL query

```

```

469 $size = count($submissions);
470 for ($i=0; $i<=$size-1; $i++) {
471     if ($this->db->dbdriver === 'postgre')
472         $query = 'UPDATE `'.$this->db->prefixTable('submissions').`_AS_t SET coefficient=_c.coeff FROM _values\n';
473     else
474         $query = 'UPDATE `'.$this->db->prefixTable('submissions').`_SET coefficient=_CASE\n';
475
476     for ($j=1000*$i; $j<1000*($i+1) && $j<$size; $j++){
477         $item = $submissions[$j];
478         if ($this->db->dbdriver === 'postgre'){
479             $query.=("assignment_id,{$item['problem']},{$item['username']},{$item['submit_id']},{$_item['coefficient']}");
480             if ($j+1<1000*($i+1) && $j+1<$size)
481                 $query.=,\n";
482         }
483         else
484             $query.=WHEN_assignment='$assignment_id'_AND_problem='{$item['problem']}'_AND_username='{$item['username']}'_
485             AND_submit_id='{$item['submit_id']}'_THEN_{$item['coefficient']}\n";
486     }
487
488     if ($this->db->dbdriver === 'postgre')
489         $query.=)_AS_c(assignment,problem,username,submit_id,coeff)\n"
490         . "WHERE_t.assignment=c.assignment_AND_t.problem=c.problem_AND_t.username=c.username_AND_t.submit_id=c.submit_id;";
491     else
492         $query.=ELSE_coefficient_\nEND_\nWHERE_assignment='$assignment_id';";
493     $this->db->query($query);
494 }
495
496 }
497 }

```

Kode B.2: HofModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Hof_model.php
5  * @author Stillmen Vallian <stillmen.v@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 class HofModel extends Model
11 {
12     protected $user_model;
13     protected $assignment_model;
14
15     public function __construct()
16     {
17         parent::__construct();
18         $this->user_model = new UserModel();
19     }
20
21
22 // -----
23
24
25 /**
26  * Get data for Hall of Fame
27  *
28  *
29  * @return mixed
30  */
31 public function get_all_final_submission()
32 {
33     //include upload only file: remove " AND file_type!='txt' AND file_type!='pdf' AND file_type!='zip' ";
34     $datas = $this->db->query("SELECT username,_CEILING(SUM(pre_score.*_coefficient/_10000))_AS_totalscore_FROM_
35         shj_submissions_WHERE_is_final=1_AND_file_type='txt'_AND_file_type!='pdf'_AND_file_type!='zip'_GROUP_BY_username_
36         ORDER_BY_totalscore_DESC")->getResultArray();
37     foreach ($datas as $key => $data) {
38         $user_id = $this->user_model->username_to_user_id($data['username']);
39         $datas[$key]['display_name'] = $this->user_model->get_user($user_id)->display_name;
40     }
41
42     return $datas;
43 }
44
45
46 /**
47  * Get details assignments & problems for selected user
48  *
49  *
50  * @return mixed
51  */
52 public function get_all_user_assignments($username)
53 {
54     $this->assignment_model = new AssignmentModel();
55     $details = $this->db->query("SELECT_assignment,_problem,_CEILING((pre_score.*_coefficient/_10000))_AS_score_FROM_
56         shj_submissions_WHERE_is_final=1_AND_username='".$username'_AND_file_type!='txt'_AND_file_type!='pdf'_AND_file_type!='
57         zip'_ORDER_BY_assignment_ASC")->getResultArray();
58     foreach ($details as $key => $detail) {
59         $assignment_id = $detail['assignment'];
60         $problem_id = $detail['problem'];
61         $details[$key]['assignment'] = $this->assignment_model->assignment_info($assignment_id)['name'];
62         $details[$key]['scoreboard'] = $this->assignment_model->assignment_info($assignment_id)['scoreboard'];
63         $details[$key]['problem'] = $this->assignment_model->problem_info($assignment_id, $problem_id)['name'];
64     }
65 }

```

```

61     }
62     return $details;
63 }
64 }
```

Kode B.3: LogsModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Logs_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 class LogsModel extends Model
11 {
12
13     public function __construct()
14     {
15         parent::__construct();
16     }
17
18     // -----
19
20
21     /**
22      * Mencatat Logs (Tabel shj_logins)
23      *
24      *
25      *
26      */
27
28     public function insert_to_logs($username, $ip_address)
29     {
30         $query = $this->db->table('logins')->get()->getResultArray();
31
32         //menghapus timestamp user yang lebih dari 24 jam
33         foreach ($query as $row)
34         {
35             $tempetime=strtotime('+24 hour', strtotime($row['timestamp']));
36             if ($tempetime < shj_now()) {
37                 # delete
38                 $builder = $this->db->table('logins');
39                 $builder->where('timestamp', $row['timestamp']);
40                 $builder->delete();
41             }
42         }
43
44         $result = $this->db->query("SELECT * FROM shj_logins WHERE username='$username' AND ip_address!='$ip_address' ORDER BY timestamp DESC")->getRow();
45         if ($result === NULL) {
46             $logins = array(
47                 'username' => $username,
48                 'ip_address' => $ip_address
49             );
50             $this->db->table('logins')->insert($logins);
51         } else {
52             $get_last_login_id = $result->login_id;
53             $logins = array(
54                 'username' => $username,
55                 'ip_address' => $ip_address,
56                 'last_24h_login_id' => $get_last_login_id
57             );
58             $this->db->table('logins')->insert($logins);
59         }
60     }
61
62
63     // -----
64
65
66     /**
67      * Get All Logs
68      *
69      * Returns an array of all logs
70      *
71      * @return mixed
72      */
73
74     public function get_all_logs()
75     {
76         return $this->db->table('logins')->orderBy('login_id', 'desc')->get()->getResultArray();
77     }
78
79
80 }
81 */
```

Kode B.4: NotificationsModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
```

```

4  * @file Notifications_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 class NotificationsModel extends Model
11 {
12
13     public function __construct()
14     {
15         parent::__construct();
16     }
17
18     // -----
19
20
21     /**
22      * Returns all notifications
23      */
24
25     public function get_all_notifications()
26     {
27         return $this->db->table('notifications')
28             ->orderBy('id', 'desc')
29             ->get()
30             ->getResultArray();
31     }
32
33
34     // -----
35
36
37     /**
38      * Returns 10 latest notifications
39      */
40
41     public function get_latest_notifications()
42     {
43         $result = $this->db->table('notifications')
44             ->limit(10)
45             ->orderBy('id', 'desc')
46             ->get()
47             ->getResultArray();
48
49         foreach($result as &$item)
50         {
51             $item['text'] = substr(trim(strip_tags($item['text'])), 0, 300);
52             $item['text'] = str_replace('&zwnj;', "\xE2\x80\x8c", $item['text']);
53             $item['text'] = html_entity_decode($item['text']);
54             $item['text'] = preg_replace('/\r?\n|\n?\r/', ' ', $item['text']);
55         }
56
57         return $result;
58     }
59
60
61     /**
62      * Add a new notification
63      */
64
65     public function add_notification($title, $text)
66     {
67         $now = shj_now_str();
68         $this->db->table('notifications')->insert(
69             array(
70                 'title' => $title,
71                 'text' => $text,
72                 'time' => $now
73             )
74         );
75
76     }
77
78     // -----
79
80     /**
81      * Update (edit) a notification
82      */
83
84     public function update_notification($id, $title, $text)
85     {
86         $this->db->table('notifications')
87             ->where('id', $id)
88             ->update(
89                 array(
90                     'title' => $title,
91                     'text' => $text
92                 )
93             );
94
95     }
96
97     // -----
98
99     /**
100      * Delete a notification
101      */
102    public function delete_notification($id)

```

```

103|     {
104|         $this->db->table('notifications')->delete(array('id' => $id));
105|
106|
107|     // -----
108|
109|
110|     /**
111|      * Returns a notification
112|      */
113|
114|     public function get_notification($notif_id)
115|     {
116|         $query = $this->db->table('notifications')->getWhere(array('id' => $notif_id));
117|         if ($query->getNumRows() != 1)
118|             return FALSE;
119|         return $query->getRowArray();
120|
121|
122|     // -----
123|
124|
125|
126|     /**
127|      * Returns true if there is a notification after $time
128|      * @todo optimize: check the ">" condition in sql query
129|      */
130|
131|     public function have_new_notification($time)
132|     {
133|         $notifs = $this->db->table('notifications')->select('time')->get()->getResultArray();
134|         foreach ($notifs as $notif) {
135|             if ($strtotime($notif['time']) > $time)
136|                 return TRUE;
137|         }
138|         return FALSE;
139|     }
140}

```

Kode B.5: QueueModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Queue_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 class QueueModel extends Model
11 {
12     protected $assignment_model;
13     protected $scoreboard_model;
14
15     public function __construct()
16     {
17         parent::__construct();
18         $this->assignment_model = new AssignmentModel();
19     }
20
21
22     // -----
23
24
25     /**
26     * Returns TRUE if one submission with $username, $assignment and $problem
27     * is already in queue (for preventing multiple submission)
28     */
29     public function in_queue ($username, $assignment, $problem)
30     {
31         $query = $this->db->table('queue')->getWhere(array('username'=>$username, 'assignment'=>$assignment, 'problem'=>$problem))
32         ;
33         return ($query->getNumRows() > 0);
34     }
35
36     // -----
37
38
39     /**
40     * Returns all the submission queue
41     */
42     public function get_queue ()
43     {
44         return $this->db->table('queue')->get()->getResultArray();
45     }
46
47
48     // -----
49
50
51     /**
52     * Empties the queue
53     */
54     public function empty_queue ()
55     {
56         return $this->db->table('queue')->emptyTable();
57     }

```



```

156        'problem' => $submission['problem'],
157        'type' => 'rejudge'
158    )));
159    // Now ready for rejudge
160 }
161
162 // -----
163
164 /**
165 * Returns the first item of the queue
166 */
167 public function get_first_item()
168 {
169     $query = $this->db->table('queue')->orderBy('id')->limit(1)->get();
170     if ($query->getNumRows() != 1){
171         return NULL;
172     }else{
173         return $query->getRowArray();
174     }
175 }
176
177 //
178
179 // -----
180
181 /**
182 * Removes an item from the queue
183 */
184 public function remove_item($username, $assignment, $problem, $submit_id)
185 {
186     $this->db->table('queue')->delete(array(
187         'submit_id' => $submit_id,
188         'username' => $username,
189         'assignment' => $assignment,
190         'problem' => $problem
191     ));
192
193     //Delete dummy submission if exec only
194     if($submit_id == 0){
195         $this->db->table('submissions')->delete(array(
196             'submit_id' => $submit_id,
197             'username' => $username,
198             'assignment' => $assignment,
199             'problem' => $problem
200         ));
201     }
202 }
203
204 //
205
206 // -----
207
208 /**
209 * Saves the result of judge in database
210 * This function is called from Queueprocess controller
211 */
212 public function save_judge_result_in_db ($submission, $type)
213 {
214
215     $arr = array(
216         'status' => $submission['status'],
217         'pre_score' => $submission['pre_score'],
218     );
219
220     if ($type === 'judge')
221     {
222         $this->db->table('submissions')->where(array(
223             'is_final' => 1,
224             'username' => $submission['username'],
225             'assignment' => $submission['assignment'],
226             'problem' => $submission['problem'],
227         ))->update(array('is_final'=>0));
228         $arr['is_final'] = 1;
229     }
230
231     $this->db->table('submissions')->where(array(
232         'submit_id' => $submission['submit_id'],
233         'username' => $submission['username'],
234         'assignment' => $submission['assignment'],
235         'problem' => $submission['problem']
236     ))->update($arr);
237
238     // update scoreboard:
239     $this->scoreboard_model = new ScoreboardModel();
240     $this->scoreboard_model->update_scoreboard($submission['assignment']);
241
242 }
243
244
245 // -----
246
247 /**
248 * Adds a dummy submission to queue for execution only
249 */
250 public function add_to_queue_exec($submit_info)
251 {
252     $query = $this->db->table('submissions')->getWhere(array(
253         'submit_id' => $submit_info['submit_id'],
254

```

```

255     'username' => $submit_info['username'],
256     'assignment' => $submit_info['assignment'],
257     'problem' => $submit_info['problem']
258   ));
259   if ($query->getNumRows() == 0){
260     $submit_info['is_final'] = 0;
261     $submit_info['status'] = 'PENDING';
262     $this->db->table('submissions')->insert($submit_info);
263
264     $this->db->table('queue')->insert(array(
265       'submit_id' => $submit_info['submit_id'],
266       'username' => $submit_info['username'],
267       'assignment' => $submit_info['assignment'],
268       'problem' => $submit_info['problem'],
269       'type' => 'exec'
270     ));
271   return TRUE;
272 }
273 else{
274   return FALSE;
275 }
276 }
277 }
```

Kode B.6: ScoreboardModel.php

```

1 <?php
2 /**
3 * Sharif Judge online judge
4 * @file Scoreboard_model.php
5 * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 class ScoreboardModel extends Model
11 {
12   protected $assignment_model;
13   protected $settings_model;
14   protected $user_model;
15
16   public function __construct()
17   {
18     parent::__construct();
19   }
20
21
22 /**
23 * Generate Scoreboard
24 *
25 * Generates scoreboard for given assignment, from Final Submissions.
26 * This function is called by update_scoreboard
27 *
28 * @param int $assignment_id
29 * @return array
30 */
31 private function _generate_scoreboard($assignment_id)
32 {
33   $this->assignment_model = new AssignmentModel();
34   $this->settings_model = new SettingsModel();
35   $assignment = $this->assignment_model->assignment_info($assignment_id);
36   $submissions = $this->db->table('submissions')->getWhere(array('is_final' => 1 , 'assignment' => $assignment_id))->getResultSet();
37   $total_score = array();
38   $penalty = array();
39   $users = array();
40   $start = strtotime($assignment['start_time']);
41   $submit_penalty = $this->settings_model->get_setting('submit_penalty');
42   $scores = array();
43   foreach ($submissions as $submission){
44
45     $pi = $this->assignment_model->problem_info($assignment_id, $submission['problem']);
46
47     $pre_score = ceil($submission['pre_score']*$pi['score']/10000);
48     if ($submission['coefficient'] === 'error')
49       $final_score = 0;
50     else
51       $final_score = ceil($pre_score*$submission['coefficient']/100);
52     $delay = strtotime($submission['time'])-$start;
53     $scores[$submission['username']][$submission['problem']]['score'] = $final_score;
54     $scores[$submission['username']][$submission['problem']]['time'] = $delay;
55
56     if ( ! isset($total_score[$submission['username']]))
57       $total_score[$submission['username']] = 0;
58     if ( ! isset($penalty[$submission['username']]))
59       $penalty[$submission['username']] = 0;
60
61     $total_score[$submission['username']] += $final_score;
62
63     $number_of_submissions = $this->db->table('submissions')->where(array(
64       'assignment' => $submission['assignment'],
65       'problem' => $submission['problem'],
66       'username' => $submission['username'],
67     ))->countAllResults();
68
69     $penalty[$submission['username']] += $delay + $number_of_submissions*$submit_penalty;
70     $users[] = $submission['username'];
71   }
}
```

```

72|     $scoreboard = array(
73|         'username' => array(),
74|         'score' => array(),
75|         'submit_penalty' => array()
76|     );
77|     $users = array_unique($users);
78|     foreach($users as $username){
79|         array_push($scoreboard['username'], $username);
80|         array_push($scoreboard['score'], $total_score[$username]);
81|         array_push($scoreboard['submit_penalty'], $penalty[$username]);
82|     }
83|     array_multisort(
84|         $scoreboard['score'], SORT_NUMERIC, SORT_DESC,
85|         $scoreboard['submit_penalty'], SORT_NUMERIC, SORT_ASC,
86|         $scoreboard['username']
87|     );
88|     return array($scores, $scoreboard);
89|
90|
91| // -----
92|
93|
94|
95| /**
96| * Update All Scoreboards
97| *
98| * Updates the cached scoreboard of all assignments.
99| * This function is called each time a user is deleted, or all submissions
100| * of a user is deleted.
101| */
102| public function update_scoreboards()
103| {
104|     $assignments = $this->db->table('assignments')->select('id')->get()->getResultArray();
105|     foreach ($assignments as $assignment){
106|         $this->update_scoreboard($assignment['id']);
107|     }
108| }
109|
110| // -----
111|
112|
113|
114| /**
115| * Update Scoreboard
116| *
117| * Updates the cached scoreboard of given assignment. Saves the html code of
118| * scoreboard table in database.
119| *
120| * This function is called after judging/rejudging a submission, and when one
121| * of these settings is changed for an assignment:
122| *
123| * TODO: Better Implementation (faster and with less db queries)
124| *
125| * - Extra Time
126| * - Start Time
127| * - Finish Time
128| * - Coefficient's Rule
129| * - Enable/Disable Scoreboard
130| *
131| * @param int $assignment_id
132| */
133| public function update_scoreboard($assignment_id)
134| {
135|     $this->assignment_model = new AssignmentModel();
136|     $this->user_model = new UserModel();
137|     // If scoreboard in not enabled, do nothing
138|     $scoreboard_enabled = $this->db->table('assignments')->select('scoreboard')->getWhere(array('id'=>$assignment_id))->getRow()
139|         ()->scoreboard;
140|     if ($assignment_id == 0 OR ! $scoreboard_enabled )
141|         return;
142|
143|     // Generate the scoreboard
144|     list ($scores, $scoreboard) = $this->_generate_scoreboard($assignment_id);
145|
146|     // Generate the scoreboard's html code
147|     // todo: Save Scoreboard as json (generate.html at client side)
148|     $all_problems = $this->assignment_model->all_problems($assignment_id);
149|     $total_score = 0;
150|     foreach($all_problems as $i)
151|         $total_score += $i['score'];
152|     $data = array(
153|         'assignment_id' => $assignment_id,
154|         'problems' => $all_problems,
155|         'total_score' => $total_score,
156|         'scores' => $scores,
157|         'scoreboard' => $scoreboard,
158|         'names' => $this->user_model->get_names(),
159|     );
160|
161|     $scoreboard_table = view('pages/scoreboard_table', $data);
162|
163|     // Minify the scoreboard's html code
164|     // $scoreboard_table = $this->output->minify($scoreboard_table, 'text/html');
165|
166|     // Save the scoreboard's html code in Database
167|     $query = $this->db->table('scoreboard')->select('assignment')->getWhere(array('assignment'=>$assignment_id));
168|     if ($query->getNumRows()==0)
169|         $this->db->table('scoreboard')->insert(array('assignment'=>$assignment_id, 'scoreboard'=>$scoreboard_table));

```

```

170     else
171         $this->db->table('scoreboard')->where('assignment', $assignment_id)->update(array('scoreboard'=>$scoreboard_table));
172     }
173
174 // -----
175
176 /**
177 * Get Cached Scoreboard
178 *
179 * Returns the cached scoreboard of given assignment as a html text
180 *
181 * @param int $assignment_id
182 * @return string
183 */
184
185 public function get_scoreboard($assignment_id)
186 {
187     $query = $this->db->table('scoreboard')->select('scoreboard')->getWhere(array('assignment'=>$assignment_id));
188
189     if ($query->getNumRows() != 1)
190         return 'Scoreboard_not_found';
191     else
192         return $query->getRow()->scoreboard;
193 }
194
195 }
196

```

Kode B.7: SettingsModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Settings_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 /**
11  * This model deals with global settings
12 */
13
14 class SettingsModel extends Model
15 {
16
17     public function __construct()
18     {
19         parent::__construct();
20     }
21
22
23 // -----
24
25
26
27     public function get_setting($key)
28     {
29         return $this->db->table('settings')->select('shj_value')->getWhere(['shj_key'=>$key])->getRow()->shj_value;
30     }
31
32
33 // -----
34
35
36     public function set_setting($key, $value)
37     {
38         $this->db->table('settings')->where('shj_key', $key)->update(array('shj_value'=>$value));
39     }
40
41
42 // -----
43
44
45     public function get_all_settings()
46     {
47         $result = $this->db->table('settings')->get()->getResultArray();
48         $settings = array();
49         foreach($result as $item)
50         {
51             $settings[$item['shj_key']] = $item['shj_value'];
52         }
53         return $settings;
54     }
55
56
57 // -----
58
59
60     public function set_settings($settings)
61     {
62         foreach ($settings as $key => $value)
63         {
64             $this->set_setting($key, $value);
65         }
66     }
67
68

```

69| }
70| }

Kode B.8: SubmitModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file Submit_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6 */
7 namespace App\Models;
8 use CodeIgniter\Model;
9
10 class SubmitModel extends Model {
11
12     protected $settings_model;
13
14     public function __construct()
15     {
16         parent::__construct();
17     }
18
19
20 // -----
21
22
23 /**
24  * Returns table row for a specific submission
25 */
26 public function get_submission($username, $assignment, $problem, $submit_id)
27 {
28     $query = $this->db->table('submissions')->getWhere(
29         array(
30             'username'=>$username,
31             'assignment'=>$assignment,
32             'problem'=>$problem,
33             'submit_id'=>$submit_id
34         )
35     );
36     if($query->getNumRows()!=1){
37         return FALSE;
38     }else{
39         return $query->getRowArray();
40     }
41 }
42
43
44 // -----
45
46
47 public function get_final_submissions($assignment_id, $user_level, $username, $page_number = NULL, $filter_user = NULL,
48     $filter_problem = NULL)
49 {
50     $this->settings_model = new SettingsModel();
51     $arr['assignment'] = $assignment_id;
52     $arr['submit_id!='] = EDITOR_SUBMIT_ID;
53     $arr['is_final'] = 1;
54     if ($user_level === 0)// students can only get final submissions of themselves
55         $arr['username']=$username;
56     elseif ($filter_user !== NULL)
57         $arr['username'] = $filter_user;
58     if ($filter_problem !== NULL)
59         $arr['problem'] = $filter_problem;
60     if ($page_number === NULL)
61         return $this->db->table('submissions')->orderBy('username_asc,problem_asc')->getWhere($arr)->getResultArray();
62     else
63     {
64         $per_page = $this->settings_model->get_setting('results_per_page_final');
65         if ($per_page == 0)
66             return $this->db->table('submissions')->orderBy('username_asc,problem_asc')->getWhere($arr)->getResultArray();
67         else
68             return $this->db->table('submissions')->orderBy('username_asc,problem_asc')->limit($per_page,($page_number-1)*
69                 $per_page)->getWhere($arr)->getResultArray();
70     }
71
72 // -----
73
74
75
76 public function get_all_submissions($assignment_id, $user_level, $username, $page_number = NULL, $filter_user = NULL,
77     $filter_problem = NULL)
78 {
79     $this->settings_model = new SettingsModel();
80     $arr['assignment']=$assignment_id;
81     $arr['submit_id!='] = EDITOR_SUBMIT_ID;
82     if ($user_level === 0)
83         $arr['username']=$username;
84     elseif ($filter_user !== NULL)
85         $arr['username'] = $filter_user;
86     if ($filter_problem !== NULL)
87         $arr['problem'] = $filter_problem;
88     if ($page_number === NULL)
89         return $this->db->table('submissions')->orderBy('submit_id','desc')->getWhere($arr)->getResultArray();
90     else
91     {

```

```

91     $per_page = $this->settings_model->get_setting('results_per_page_all');
92     if ($per_page == 0)
93         return $this->db->table('submissions')->orderBy('submit_id','desc')->getWhere($arr)->getResultArray();
94     else
95         return $this->db->table('submissions')->orderBy('submit_id','desc')->limit($per_page,($page_number-1)*$per_page)->
96             getWhere($arr)->getResultArray();
97 }
98
99 // -----
100
101
102 public function count_final_submissions($assignment_id, $user_level, $username, $filter_user = NULL, $filter_problem = NULL)
103 {
104     $arr['assignment'] = $assignment_id;
105     $arr['submit_id_!='] = EDITOR_SUBMIT_ID;
106     $arr['is_final'] = 1;
107     if ($user_level === 0)
108         $arr['username']=$username;
109     elseif ($filter_user !== NULL)
110         $arr['username'] = $filter_user;
111     if ($filter_problem !== NULL)
112         $arr['problem'] = $filter_problem;
113     return $this->db->table('submissions')->where($arr)->countAllResults();
114 }
115
116
117 // -----
118
119
120 public function count_all_submissions($assignment_id, $user_level, $username, $filter_user = NULL, $filter_problem = NULL)
121 {
122     $arr['assignment']=$assignment_id;
123     $arr['submit_id_!='] = EDITOR_SUBMIT_ID;
124     if ($user_level === 0)
125         $arr['username']=$username;
126     elseif ($filter_user !== NULL)
127         $arr['username'] = $filter_user;
128     if ($filter_problem !== NULL)
129         $arr['problem'] = $filter_problem;
130     return $this->db->table('submissions')->where($arr)->countAllResults();
131 }
132
133
134 // -----
135
136
137 public function set_final_submission($username, $assignment, $problem, $submit_id)
138 {
139
140     $this->db->table('submissions')->where(array(
141         'is_final' => 1,
142         'username' => $username,
143         'assignment' => $assignment,
144         'problem' => $problem,
145     ))->update(array('is_final'=>0));
146
147     $this->db->table('submissions')->where(array(
148         'username' => $username,
149         'assignment' => $assignment,
150         'problem' => $problem,
151         'submit_id' => $submit_id,
152     ))->update(array('is_final'=>1));
153
154     return TRUE;
155 }
156
157
158 // -----
159
160
161 /**
162 * add the result of an "upload only" submit to the database
163 */
164
165 public function add_upload_only($submit_info)
166 {
167
168     $this->db->table('submissions')->where(array(
169         'is_final' => 1,
170         'username' => $submit_info['username'],
171         'assignment' => $submit_info['assignment'],
172         'problem' => $submit_info['problem'],
173     ))->update(array('is_final'=>0));
174
175     $submit_info['is_final'] = 1;
176     $submit_info['status'] = 'Uploaded';
177
178     $this->db->table('submissions')->insert($submit_info);
179
180 }
181
182
183 }

```

Kode B.9: User.php

```

1| <?php
2| /**
3| * SharIF Judge online judge
4| * @file User_model.php
5| * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6| */
7|
8| namespace App\Models;
9| use CodeIgniter\Model;
10| use Config\Database;
11|
12| class User extends Model
13| {
14|     public $username;
15|     public $selected_assignment;
16|     public $level;
17|     public $email;
18|     protected $session;
19|     protected $db;
20|
21|     public function __construct()
22|     {
23|         parent::__construct();
24|         $this->session = session();
25|         $this->db = Database::connect();
26|         $this->username = $this->session->get('username');
27|         if ($this->username === NULL)
28|             return;
29|
30|         $user = $this->db->table('users')
31|             ->select('selected_assignment, role, email')
32|             ->where(['username' => $this->username])
33|             ->get()->getRow();
34|
35|         $this->email = $user->email;
36|
37|         $query = $this->db->table('assignments')->getWhere(['id' => $user->selected_assignment]);
38|         if ($query->getNumRows() != 1)
39|             $this->selected_assignment = array(
40|                 'id' => 0,
41|                 'name' => 'Not Selected',
42|                 'finish_time' => '0',
43|                 'extra_time' => '0',
44|                 'problems' => 0
45|             );
46|         else
47|             $this->selected_assignment = $query->getRowArray();
48|
49|         switch ($user->role)
50|         {
51|             case 'admin': $this->level = 3; break;
52|             case 'head_instructor': $this->level = 2; break;
53|             case 'instructor': $this->level = 1; break;
54|             case 'student': $this->level = 0; break;
55|         }
56|
57|
58| // -----
59|
60|
61| /**
62| * Select Assignment
63| *
64| * Sets selected assignment for $username
65| *
66| * @param $assignment_id
67| */
68| public function select_assignment($assignment_id)
69| {
70|     $this->db->table('users')->where('username', $this->username)->update(['selected_assignment'=>$assignment_id]);
71|
72| }
73|
74| // -----
75|
76|
77| /**
78| * Save Widget Positions
79| *
80| * Updates position of dashboard widgets in database
81| *
82| * @param $positions
83| */
84| public function save_widget_positions($positions)
85| {
86|     $this->db->table('users')
87|         ->where('username', $this->username)
88|         ->update(['dashboard_widget_positions'=>$positions]);
89| }
90|
91|
92| // -----
93|
94|
95| /**
96| * Get Widget Positions
97| *
98| * Returns positions of dashboard widgets from database
99|

```

```

100    *
101   * @param none
102   * @return mixed
103   */
104  public function get_widget_positions()
105  {
106      $tab = $this->db->table('users')
107      ->select('dashboard_widget_positions')
108      ->getWhere(['username' => $this->username])
109      ->getRow()
110      ->dashboard_widget_positions;
111
112      return json_decode(
113          $tab,
114          TRUE
115      );
116  }
117 }
118 }
```

Kode B.10: UserModel.php

```

1 <?php
2 /**
3  * SharIF Judge online judge
4  * @file User_model.php
5  * @author Mohammad Javad Naderi <mjnaderi@gmail.com>
6  */
7 namespace App\Models;
8 use CodeIgniter\Model;
9 use Dapphp\Radius\Radius;
10 use App\Models\ScoreboardModel;
11 use App\Models\SettingsModel;
12
13 class UserModel extends Model
14 {
15     protected $validation;
16     protected $config;
17     protected $email;
18     protected $settings_model;
19     protected $scoreboard_model;
20     protected $session;
21     protected $response;
22
23     public function __construct()
24     {
25         parent::__construct();
26         $this->validation = \Config\Services::validation();
27         $this->config = config('Secrets');
28         $this->settings_model = new SettingsModel();
29         $this->scoreboard_model = new ScoreboardModel();
30         $this->email = \Config\Services::email();
31     }
32     // -----
33
34 /**
35  * Have User
36  *
37  * Returns TRUE if there is a user with username $username in database
38  *
39  * @param $username
40  * @return bool
41  */
42 public function have_user($username)
43 {
44     $query = $this->db->table('users')->getWhere(['username'=>$username]);
45     if ($query->getNumRows() == 0)
46         return FALSE;
47     if ($username === $query->getRow()->username) // needed (because of utf8_general_ci [ci=case insensitive])
48         return TRUE;
49     return FALSE;
50 }
51
52 // -----
53
54
55 /**
56  * User ID to Username
57  *
58  * Converts user id to username (returns FALSE if user does not exist)
59  *
60  * @param $user_id
61  * @return bool
62  */
63
64 public function user_id_to_username($user_id)
65 {
66     if( ! is_numeric($user_id))
67         return FALSE;
68     $query = $this->db->table('users')->select('username')->getWhere(['id'=>$user_id]);
69     if ($query->getNumRows() == 0)
70         return FALSE;
71     return $query->getRow()->username;
72 }
73
74 // -----
75
76 }
```

```

77 /**
78 * Username to User ID
79 *
80 * Converts username to user id (returns FALSE if user does not exist)
81 *
82 * @param $username
83 * @return bool
84 */
85 public function username_to_user_id($username)
86 {
87     $query = $this->db->table('users')->select('id')->getWhere(['username'=>$username]);
88     if ($query->getNumRows() == 0)
89         return FALSE;
90     return $query->getRow()->id;
91 }
92
93
94 /**
95 -----
96
97 /**
98 * Have Email
99 *
100 * Returns TRUE if a user (except $username) with given email exists
101 *
102 * @param $email
103 * @param bool $username
104 * @return bool
105 */
106 public function have_email($email, $username = FALSE)
107 {
108     $query = $this->db->table('users')->getWhere(['email'=>$email]);
109     if ($query->getNumRows() >= 1){
110         if($username != FALSE && $query->getRow()->username == $username)
111             return FALSE;
112         else
113             return TRUE;
114     }
115     return FALSE;
116 }
117
118 /**
119 -----
120
121
122 /**
123 * Add User
124 *
125 * Adds a single user
126 *
127 * @param $username
128 * @param $email
129 * @param $password
130 * @param $role
131 * @return bool|string
132 */
133 public function add_user($username, $email, $display_name, $password, $role)
134 {
135     if ( ! ctype_alnum($username) )
136         return 'Username_may_only_contain_alpha-numeric_characters.';
137     if (strlen($username) < 3 OR strlen($username) > 20 OR strlen($password) < 6 OR strlen($password) > 200)
138         return 'Username_or_password_length_error.';
139     if ($this->have_user($username))
140         return 'User_with_this_username_exists.';
141     if ($this->have_email($email))
142         return 'User_with_this_email_exists.';
143     if (strtolower($username) !== $username)
144         return 'Username_must_be_lowercase.';
145     $roles = array('admin', 'head_instructor', 'instructor', 'student');
146     if ( ! in_array($role, $roles))
147         return 'Users_role_is_not_valid.';
148
149     $user=[
150         'username' => $username,
151         'email' => $email,
152         'display_name' => $display_name,
153         'password' => password_hash($password,PASSWORD_BCRYPT),
154         'role' => $role
155     ];
156     $this->db->table('users')->insert($user);
157     return TRUE; //success
158 }
159
160
161 /**
162 -----
163
164 /**
165 * Add Users
166 *
167 * Adds multiple users
168 *
169 * @param $text
170 * @param $send_mail
171 * @param $delay
172 * @return array
173 */
174 public function add_users($text, $send_mail, $delay)
175

```

```

176{
177
178    $lines = preg_split('/\r?\n|\n?\r/', $text);
179    $users_ok = array();
180    $users_error = array();
181
182    // loop over lines of $text :
183    foreach ($lines as $line)
184    {
185        $line = trim($line);
186
187        if (strlen($line) == 0 OR $line[0] == '#')
188            continue; //ignore comments and empty lines
189
190        $parts = preg_split('/,+/', $line);
191        if (count($parts) == 5){
192            if (strtolower(substr($parts[3], 0, 6)) == 'random')
193            {
194                // generate random password
195                $len = trim(substr($parts[3], 6), '[]');
196                if (is_numeric($len)){
197                    $this->load->helper('string');
198                    $parts[3] = shj_random_password($len);
199                }
200            }
201
202            $result = $this->add_user($parts[0], $parts[1], $parts[2], $parts[3], $parts[4]);
203
204            if ($result == TRUE)
205                array_push($users_ok, array($parts[0], $parts[1], $parts[2], $parts[3], $parts[4]));
206            else
207                array_push($users_error, array($parts[0], $parts[1], $parts[2], $parts[3], $parts[4], $result));
208
209        else{
210            array_push($users_error, array($parts[0], $parts[1], $parts[2], $parts[3], $parts[4], 'Wrong_Format'));
211            continue; //ignore lines that not contain 5 parts
212        }
213    } // end of loop
214
215    if ($send_email)
216    {
217        // sending usernames and passwords by email
218        $config = [
219            'mailtype' => 'html',
220            'charset'  => 'iso-8859-1'
221        ];
222        $this->email->initialize($this->config->shj_mail);
223        $this->email->setNewline("\r\n");
224        $count_users = count($users_ok);
225        $counter = 0;
226        foreach ($users_ok as $user)
227        {
228            $counter++;
229            $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'));
230            $this->email->setTo($user[1]);
231            $this->email->setSubject('SharIF_Judge_Username_and_Password');
232            $text = $this->settings_model->get_setting('add_user_mail');
233            $text = str_replace('{SITE_URL}', base_url(), $text);
234            $text = str_replace('{ROLE}', $user[4], $text);
235            $text = str_replace('{USERNAME}', $user[0], $text);
236            $text = str_replace('{PASSWORD}', htmlspecialchars($user[3]), $text);
237            $text = str_replace('{LOGIN_URL}', base_url(), $text);
238            $this->email->setMessage($text);
239            $this->email->send();
240            if ($counter < $count_users)
241                sleep($delay);
242        }
243    }
244
245    return array($users_ok, $users_error);
246
247}
248
249
250// -----
251
252
253< /**
254 * Delete User
255 *
256 * Deletes a user with given user id
257 * Returns TRUE (success) or FALSE (failure)
258 *
259 * @param $user_id
260 * @return bool
261 */
262
263public function delete_user($user_id)
264{
265    $this->db->transStart();
266
267    $username = $this->user_id_to_username($user_id);
268    if ($username === FALSE)
269        return FALSE;
270    $this->db->table('users')->delete(['id'=>$user_id]);
271    $this->db->table('submissions')->delete(['username' => $username]);
272    // each time we delete a user, we should update all scoreboards
273    $this->scoreboard_model->update_scoreboards();

```

```

274     $this->db->transComplete();
275
276     if ($this->db->transStatus()) {
277         // Delete submitted files
278         shell_exec("cd_{$this->settings_model->get_setting('assignments_root')};_rm_-r_/*/{$username};");
279         return TRUE; // success
280     }
281     return FALSE; // failure
282 }
283
284
285 // -----
286
287 /**
288 * Delete Submissions
289 *
290 * Deletes all submissions of user with given user id
291 * Returns TRUE (success) or FALSE (failure)
292 *
293 * @param $user_id
294 * @return bool
295 */
296 public function delete_submissions($user_id)
297 {
298     $this->db->transStart();
299
300     $username = $this->user_id_to_username($user_id);
301     if ($username === FALSE)
302         return FALSE;
303     // delete all submissions from database
304     $this->db->table('submissions')->delete(['username'=>$username]);
305     // each time we delete a user's submissions, we should update all scoreboards
306     $this->scoreboard_model->update_scoreboards();
307
308     $this->db->transComplete();
309
310     if ($this->db->transStatus()) {
311         // delete all submitted files
312         shell_exec("cd_{$this->settings_model->get_setting('assignments_root')};_rm_-r_/*/{$username};");
313         return TRUE; // success
314     }
315
316     return FALSE; // failure
317 }
318
319
320 // -----
321
322
323 /**
324 * Validate User
325 *
326 * Returns TRUE if given username and password is valid for login
327 *
328 * @param $username
329 * @param $password
330 * @return bool
331 */
332 public function validate_user($username, $password)
333 {
334     $query = $this->db->table('users')->getWhere(['username' => $username]);
335     if ($query->getNumRows() != 1)
336         return FALSE;
337     if ($query->getRow()->username !== $username) // needed (because of utf8_general_ci [ci=case insensitive])
338         return FALSE;
339     if (password_verify($password, $query->getRow()->password))
340         return TRUE;
341
342     if($this->config->shj_authenticate == 'radius') {
343         $client = new Radius();
344         $client->setServer($this->config->shj_radius['server']); // RADIUS server address
345         ->setSecret($this->config->shj_radius['secret']);
346         if($client->accessRequest($username, $password))
347             return TRUE;
348     } else if($this->config->shj_authenticate == 'ldap') {
349         try {
350             $ldap = new \Adldap\Adldap();
351             $ldap->addProvider($this->config->shj_ldap);
352
353             $guard = $ldap->getDefaultProvider()->getGuard();
354             return $guard->attempt($username, $password);
355         } catch (\Exception $e) {
356             return FALSE;
357         }
358     }
359     return FALSE;
360 }
361
362
363
364 // -----
365
366
367 /**
368 * Selected Assignment
369 *
370 * Returns selected assignment by given username
371 * @param $username
372 * @return mixed

```

```

373  /*
374  public function selected_assignment($username)
375  {
376      $this->session = session();
377      $query = $this->db->table('users')->select('selected_assignment')->getWhere(['username'=>$username]);
378      if ($query->getNumRows() != 1){//logout
379          $this->session->destroy();
380          return redirect()->to('login');
381      }
382      return $query->getRow()->selected_assignment;
383  }
384
385  // -----
386
387
388 /**
389 * Get Display Name
390 *
391 * Returns name of the user with given username
392 *
393 * @return array
394 */
395 public function get_names()
396 {
397     $query = $this->db->table('users')->select('username,_display_name')->get();
398     $tmp = $query->getResultArray();
399     $result = array();
400     foreach ($tmp as $row)
401     {
402         $result[$row['username']] = $row['display_name'];
403     }
404     return $result;
405 }
406
407 // -----
408
409 /**
410 * Update Profile
411 *
412 * Updates User Profile (Name, Email, Password, Role)
413 *
414 * @param $user_id
415 * @return bool
416 */
417 public function update_profile($user_id,$email,$password,$role,$display_name)
418 {
419     $query = $this->db->table('users')->getWhere(['id'=>$user_id]);
420     if ($query->getNumRows() != 1)
421         return FALSE;
422     $the_user = $query->getRow();
423     $username = $the_user->username;
424
425     $display_name = $display_name;
426     $locked = $this->settings_model->get_setting('lock_student_display_name');
427
428     if ($locked == 1) {
429         $display_name = $the_user->display_name;
430     }
431
432     $user=array(
433         'display_name' => $display_name,
434         'email' => $email
435     );
436
437     // if a role is provided, change the role
438     // (only admins are able to provide a role)
439     if ($role !== NULL)
440         $user['role'] = $role;
441
442     // if a password is provided, change the password:
443     if ($password != '')
444         $user['password'] = password_hash($password, PASSWORD_BCRYPT);
445
446     $this->db->table('users')->where('username', $username)->update($user);
447
448 }
449
450
451 // -----
452
453
454 /**
455 * Send Password Reset Mail
456 *
457 * Generates a password reset key and sends an email containing the link
458 * for resetting password (in case of password lost)
459 *
460 * @param $email
461 */
462 public function send_password_reset_mail($email)
463 {
464     // exit if $email is invalid:
465     if ( !$this->have_email($email) )
466         return;
467
468     // generate a random password reset key:
469     helper('string');
470     $passchange_key = random_string('alnum', 50);
471

```

```

472 // save the key in users table:
473 $now = shj_now();
474 $this->db->table('users')->where('email', $email)->update(array('passchange_key'=>$passchange_key, 'passchange_time'=>date
475 ('Y-m-d_H:i:s', $now)));
476
477 // send the email:
478 $this->response = response();
479
480 $this->email->initialize($this->config->shj_mail);
481 $this->email->setNewline("\r\n");
482 $this->email->setFrom($this->settings_model->get_setting('mail_from'), $this->settings_model->get_setting('mail_from_name'
483 ));
484 $this->email->setTo($email);
485 $this->email->setSubject('Password Reset');
486 $text = $this->settings_model->get_setting('reset_password_mail');
487 $text = str_replace('{SITE_URL}', base_url(), $text);
488 $text = str_replace('{RESET_LINK}', site_url('login/reset/'.$passchange_key), $text);
489 $text = str_replace('{VALID_TIME}', '1_hour', $text); // links are valid for 1 hour
490 $this->email->setMessage($text);
491 if(!$this->email->send()){
492     dd($this->email->printDebugger());
493 }
494
495 // -----
496
497
498 /**
499 * Password Reset Key Is Valid
500 *
501 * Returns TRUE if the given password reset key is valid
502 * And returns an error message if key is invalid
503 *
504 * @param $passchange_key
505 * @return bool|string
506 */
507 public function passchange_is_valid($passchange_key)
508 {
509     $query = $this->db->table('users')->select('passchange_time')->getWhere(['passchange_key'=>$passchange_key]);
510     if ($query->getNumRows() != 1)
511         return 'Invalid_password_reset_link.';
512     $time = strtotime($query->getRow()->passchange_time);
513     $now = shj_now();
514     if ($now-$time > 3600 OR $now-$time < 0) // reset link is valid for 1 hour
515         return 'The_link_is_expired.';
516     return TRUE;
517 }
518
519
520 // -----
521
522
523 /**
524 * Reset Password
525 *
526 * Resets password for given password reset key (in case of lost password)
527 *
528 * @param $passchange_key
529 * @param $newpassword
530 * @return bool
531 */
532 public function reset_password($passchange_key, $newpassword)
533 {
534     $query = $this->db->table('users')->getWhere(array('passchange_key'=>$passchange_key));
535     if ($query->getNumRows() != 1)
536         return FALSE; //failure
537     $this->db->table('users')->where('username', $query->getRow()->username)->update(array('passchange_key'=>'', 'password' =>
538         password_hash($newpassword,PASSWORD_BCRYPT)));
539     return TRUE; //success
540 }
541
542 // -----
543
544
545 /**
546 * Get All Users
547 *
548 * Returns an array of all users (for Users page)
549 *
550 * @return mixed
551 */
552 public function get_all_users()
553 {
554     return $this->db->table('users')->orderBy('role', 'asc')->orderBy('id')->get()->getResultArray();
555 }
556
557
558 // -----
559
560
561 /**
562 * Get User
563 *
564 * Returns database row for given user id
565 *
566 * @param $user_id
567 */

```

```
568     * @return bool
569     */
570     public function get_user($user_id)
571     {
572         $query = $this->db->table('users')->getWhere(array('id'=>$user_id));
573         if ($query->getNumRows() != 1)
574             return FALSE;
575         return $query->getRow();
576     }
577
578     // -----
579
580
581 /**
582 * Update Login Time
583 *
584 * Updates First Login Time and Last Login Time for given username
585 *
586 */
587
588 public function update_login_time($username)
589 {
590     helper('shj_helper');
591     $now = shj_now_str();
592
593     $first_login = $this->db->table('users')->select('first_login_time')->getWhere(array('username'=>$username))->getRow() ->
594         first_login_time;
595     if ($first_login === NULL)
596         $this->db->table('users')->where('username', $username)->update(array('first_login_time'=>$now));
597
598     $this->db->table('users')->where('username', $username)->update(array('last_login_time'=>$now));
599
600 }
601
602 // -----
```

LAMPIRAN C

KODE PROGRAM *VIEWS*

Kode C.1: add_assignment.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: add_assignment.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?><?= $edit ? 'fa-edit' : 'fa-plus-square' ?><?= $this->endSection() ?>
8 <?= $this->section('title') ?><?= $edit ? 'Edit' : 'Add' ?><?= $this->endSection() ?>
9 <?= $this->section('head_title') ?><?= $edit ? 'Edit' : 'Add' ?><?= $this->endSection() ?>
10
11
12 <?= $this->section('other_assets') ?>
13 <script type='text/javascript' src="= _base_url('assets/js/taboverride.min.js') ?&gt;"&gt;&lt;/script&gt;
14 &lt;script&gt;
15     $(document).ready(function(){
16         tabOverride.set(document.getElementsByName('textarea'));
17         $switch = false;
18         $("textarea").keyup(function (e) {
19             if (e.which==27){
20                 tabOverride.set(document.getElementsByName('textarea'),$switch);
21                 $switch = !$switch;
22             }
23         });
24     });
25 &lt;/script&gt;
26 &lt;script type="text/javascript" src="<?= _base_url('assets/js/jquery-ui-timepicker-addon.js') ?&gt;"&gt;&lt;/script&gt;
27 &lt;script&gt;
28     shj.num_of_problems=&lt;?= count($problems) ?&gt;;
29     shj.row='&lt;tr&gt;&lt;td&gt;PID&lt;/td&gt;\n
30     &lt;td&gt;&lt;input_aria-label="Problem_Name" type="text" name="name[]" class="sharif_input_short" value="Problem_"/&gt;&lt;/td&gt;\n
31     &lt;td&gt;&lt;input_aria-label="Score" type="text" name="score[]" class="sharif_input_tiny2" value="100"/&gt;&lt;/td&gt;\n
32     &lt;td&gt;&lt;input_aria-label="Time_Limit_for_C" type="text" name="c_time_limit[]" class="sharif_input_tiny2" value="500"/&gt;&lt;/td&gt;\n
33     &lt;td&gt;&lt;input_aria-label="Time_Limit_for_Python" type="text" name="python_time_limit[]" class="sharif_input_tiny2" value
34     ="1500"/&gt;&lt;/td&gt;\n
35     &lt;td&gt;&lt;input_aria-label="Time_Limit_for_Java" type="text" name="java_time_limit[]" class="sharif_input_tiny2" value="2000"/&gt;&lt;/td&gt;
36     &gt;\n
37     &lt;td&gt;&lt;input_aria-label="Memory_Limit" type="text" name="memory_limit[]" class="sharif_input_tiny" value="50000"/&gt;&lt;/td&gt;\n
38     &lt;td&gt;&lt;input_aria-label="Allowed_Languages" type="text" name="languages[]" class="sharif_input_short2" value="C,C++,Python_2,
39     Python_3,Java"/&gt;&lt;/td&gt;\n
40     &lt;td&gt;&lt;input_aria-label="Diff_Command" type="text" name="diff_cmd[]" class="sharif_input_tiny" value="diff"/&gt;&lt;/td&gt;\n
41     &lt;td&gt;&lt;input_aria-label="Diff_Argument" type="text" name="diff_arg[]" class="sharif_input_tiny" value=".bB"/&gt;&lt;/td&gt;\n
42     &lt;td&gt;&lt;input_aria-label="Upload_Only" type="checkbox" name="is_upload_only[]" class="check" value="PID"/&gt;&lt;/td&gt;\n
43     &lt;td&gt;&lt;button class="delete_problem" type="button" aria-label="Delete_Problem"&gt;&lt;i class="fa fa-times-circle fa-lg_color1_pointer
44     "&gt;&lt;/i&gt;&lt;/button&gt;&lt;/td&gt;\n
45 &lt;/tr&gt;';
46     $(document).ready(function(){
47         $('#add').click(function(){
48             $('#problems_table&gt;tbody').append(shj.row.replace(/PID/g, (shj.num_of_problems+1)));
49             shj.num_of_problems++;
50             $('#nop').attr('value', shj.num_of_problems);
51         });
52         $("#form_a_archived_assignment").click(function(){
53             if ($("#form_a_archived_assignment").is(':checked')) {
54                 $("#start_time").val('1970-01-02_00:00:00');
55                 $("#finish_time").val('2038-01-18_00:00:00');
56                 $("#form_extra_time").val('0');
57             } else{
58                 $("#start_time").val('');
59                 $("#finish_time").val('');
60                 $("#form_extra_time").val('');
61             });
62             $(document).on('click', '.delete_problem', function(){
63                 if (shj.num_of_problems==1) return;
64                 var row = $(this).parents('tr');
65                 row.remove();
66                 var i = 0;
67                 $('#problems_table&gt;tbody').children('tr').each(function(){
68                     i++;
69                     $(this).children(':first').html(i);
70                     $(this).find('[type="checkbox"]').attr('value',i);
71                 });
72                 shj.num_of_problems--;
73                 $('#nop').attr('value',shj.num_of_problems);
74             });
75         });
76     });
77 </pre

```

```

72    });
73    $('#start_time').datetimepicker({
74        timeFormat: 'HH:mm:ss'
75    });
76    $('#finish_time').datetimepicker({
77        timeFormat: 'HH:mm:ss'
78    });
79 });
80 </script>
81 <?= $this->endSection() ?>
82
83
84 <?= $this->section('title_menu') ?>
85 <span class="title_menu_item">
86     <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md" target="_blank"><i class="fa fa-question-circle color1"></i> Help</a>
87 </span>
88 <?= $this->endSection() ?>
89
90
91
92 <?= $this->section('main_content') ?>
93 <?php $msgclasses = ['success' => 'shj_g', 'notice' => 'shj_o', 'error' => 'shj_r'] ?>
94 <?php foreach($messages as $message): ?>
95     <p class="<?= $msgclasses[$message->type] ?>"><? $message->text ?></p>
96 <?php endforeach ?>
97
98 <?php if($edit): ?>
99 <p>
100     <i class="fa fa-info-circle fa-lg color8"></i> If you don't want to change tests or pdf file, just do not upload its file.
101 </p>
102 <?php endif ?>
103
104 <?= form_open_multipart($edit ? '_assignments/edit/' . $edit_assignment['id'] : '_assignments/add') ?>
105 <div class="panel_left">
106     <input type="hidden" name="number_of_problems" id="nop" value="<?= $edit ? $edit_assignment['problems'] : count($problems) ?>" />
107     <p class="input_p">
108         <label for="form_a_name">Assignment Name</label>
109         <input id="form_a_name" type="text" name="assignment_name" class="sharif_input_medium" value="<?= $edit ? $edit_assignment['name'] : set_value('assignment_name') ?>" />
110         <div class="shj_error"><?=$validationError->hasError('assignment_name') ?> $validationError->getError('assignment_name') : '' ?></div>
111     </p>
112     <p class="input_p">
113         <label for="start_time">Start Time</label>
114         <input id="start_time" type="text" name="start_time" class="sharif_input_medium" value="<?= $edit ? date('m/d/Y H:i:s', strtotime($edit_assignment['start_time'])) : set_value('start_time') ?>" />
115         <div class="shj_error"><?=$validationError->hasError('start_time') ?> $validationError->getError('start_time') : '' ?></div>
116     </p>
117     <p class="input_p">
118         <label for="finish_time">Finish Time</label>
119         <input id="finish_time" type="text" name="finish_time" class="sharif_input_medium" value="<?= $edit ? date('m/d/Y H:i:s', strtotime($edit_assignment['finish_time'])) : set_value('finish_time') ?>" />
120         <div class="shj_error"><?=$validationError->hasError('finish_time') ?> $validationError->getError('finish_time') : '' ?></div>
121     </p>
122     <p class="input_p_clear">
123         <label for="form_extra_time">
124             Extra Time (minutes)<br>
125             <span class="form_comment">Extra time for late submissions.</span>
126         </label>
127     </p>
128     <input id="form_extra_time" type="text" name="extra_time" class="sharif_input_medium" value="<?= $edit ? $twig->extra_time_formatter($edit_assignment['extra_time']) : set_value('extra_time') ?>" />
129     <div class="shj_error"><?=$validationError->hasError('extra_time') ?> $validationError->getError('extra_time') : '' ?></div>
130     </p>
131     <p class="input_p_clear">
132         <label for="form_participants">Participants<br>
133             <span class="form_comment">Enter username of participants here (comma separated).<br>
134             Only these users are able to submit. You can use keyword "ALL".</span>
135             <span class="form_comment_clear">Press "esc" to enable/disable tab indent</span>
136         </label>
137         <textarea id="form_participants" name="participants" rows="5" class="sharif_input_medium"><?= $edit ? $edit_assignment['participants'] : set_value('participants', 'ALL') ?></textarea>
138     </p>
139     <p class="input_p_clear">
140         <label for="form_tests_desc">Tests and Descriptions (zip file)<br>
141             <span class="form_comment">
142                 <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/tests_structure.md" target="_blank">Use this structure</a>
143             </span>
144         </label>
145         <input id="form_tests_desc" type="file" name="tests_desc" class="sharif_input_medium"/>
146     </p>
147     <p class="input_p_clear">
148         <label for="form_pdf">PDF File<br>
149             <span class="form_comment">
150                 PDF file of Assignment
151             </span>
152         </label>
153         <input id="form_pdf" type="file" name="pdf" class="sharif_input_medium"/>
154     </p>
155 </div>
156 <div class="panel_right">
157     <p class="input_p">

```

```

158 <input_id="form_a_open"_type="checkbox"_name="open"_value="1"_{<=?_$_edit_?_($edit_assignment['open'])_?_checked'_&:_?}>/>
159 <label_for="form_a_open"_class="default">Open</label>
160 <span_class="form_comment_space-left">Open_or_close_this_assignment</span>
161 <div_class="shj_error"><=?_$_validationError->hasError('open')_?_$_validationError->getError('open')_&:_?></div>
162 </p>
163 <p_class="input_p">
164 <input_id="form_a_scoreboard"_type="checkbox"_name="scoreboard"_value="1"_{<=?_$_edit_?_($edit_assignment['scoreboard'])_?_checked'_&:_?}>/>
165 <label_for="form_a_scoreboard"_class="default">Scoreboard</label>
166 <span_class="form_comment_space-left">Check_this_to_enable_scoreboard</span>
167 <div_class="shj_error"><=?_$_validationError->hasError('scoreboard')_?_$_validationError->getError('scoreboard')_&:_?></div>
168 </p>
169 <p_class="input_p">
170 <input_id="form_a_javaexceptions"_type="checkbox"_name="javaexceptions"_value="1"_{<=?_$_edit_?_($edit_assignment['javaexceptions'])_?_checked'_&:_?}>/>
171 <label_for="form_a_javaexceptions"_class="default">Java_Exceptions</label>
172 <span_class="form_comment_space-left">Check_this_to_show_Java_exceptions_to_users</span>
173 <div_class="shj_error"><=?_$_validationError->hasError('javaexceptions')_?_$_validationError->getError('javaexceptions')_&:_?></div>
174 </p>
175 <p_class="input_p">
176 <input_id="form_a_archived_assignment"_type="checkbox"_name="archived_assignment"_value="1"_{<=?_$_edit_?_($edit_assignment['archived_assignment'])_?_checked'_&:_?}>/>
177 <label_for="form_a_archived_assignment"_class="default">Archived_Assignment</label>
178 <span_class="form_comment_space-left">Check_this_to_make_an_archived_assignment</span>
179 <div_class="shj_error"><=?_$_validationError->hasError('archived_assignment')_?_$_validationError->getError('archived_assignment')_&:_?></div>
180 </p>
181 <p_class="input_p">
182 <label_for="form_late_rule">Coefficient_rule_{<a_target="_blank"_href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#coefficient-rule"></a>}</label><br>
183 <span_class="form_comment_medium_clear" style="display: block;">PHP_script_without_&lt;?php_&gt;_tags</span>
184 <span_class="form_comment_clear">Press_&lt;esc_&gt;_to_enable/disable_tabindent</span><br>
185 <textarea_id="form_late_rule"_name="late_rule"_rows="20"_class="sharif_input_add_text"><=?_$_edit_?_$_edit_assignment['late_rule']_>_><set_value('late_rule',$_default_late_rule)></textarea>
186 <div_class="shj_error"><=?_$_validationError->hasError('late_rule')_?_$_validationError->getError('late_rule')_&:_?></div>
187 </p>
188 </div>
189 <p_class="input_p" id="add_problems">Problems<button_type="button" id="add" aria-label="Add_Problem"><i_class="fa_fa-plus-circle_fa-lg_color1_pointer"></i></button>
190 <table_id="problems_table">
191 <thead>
192 <tr>
193 <th_rowspan="2"></th>
194 <th_rowspan="2">Name</th>
195 <th_rowspan="2">Score</th>
196 <th_colspan="3" style="border-bottom: 1px_solid_#BDBDBD">Time_Limit_(ms)</th>
197 <th_rowspan="2">Memory<br>Limit_(KB)</th>
198 <th_rowspan="2">Allowed<br>Languages_{<a_aria-label="Link_Help_For_Languages" target="_blank" href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#allowed-languages"></a>}</th>
199 <th_rowspan="2">Diff<br>Command_{<a_aria-label="Link_Help_For_Diff_Command" target="_blank" href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#diff-command"></a>}</th>
200 <th_rowspan="2">Diff<br>Argument_{<a_aria-label="Link_Help_For_Diff_Argument" target="_blank" href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#diff-arguments"></a>}</th>
201 <th_rowspan="2">Upload<br>Only_{<a_aria-label="Link_Help_For_Upload_Only" target="_blank" href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#upload-only"></a>}</th>
202 <th_rowspan="2"></th>
203 </tr>
204 <tr>
205 <th>C/C++</th><th>Python</th><th>Java</th>
206 </tr>
207 </thead>
208 <tbody>
209 <?php foreach_($problems_as_$problem) :?>
210 <tr>
211 <td><=?_$_problem['id']_?></td>
212 <td><input_aria-label="Problem_Name" type="text" name="name[]" class="sharif_input_short" value="<=?_$_problem['name']_?>"></td>
213 <td><input_aria-label="Score" type="text" name="score[]" class="sharif_input_tiny2" value="<=?_$_problem['score']_?>"></td>
214 <td><input_aria-label="Time_Limit_for_C" type="text" name="c_time_limit[]" class="sharif_input_tiny2" value="<=?_$_problem['c_time_limit']_?>"></td>
215 <td><input_aria-label="Time_Limit_for_Python" type="text" name="python_time_limit[]" class="sharif_input_tiny2" value="<=?_$_problem['python_time_limit']_?>"></td>
216 <td><input_aria-label="Time_Limit_for_Java" type="text" name="java_time_limit[]" class="sharif_input_tiny2" value="<=?_$_problem['java_time_limit']_?>"></td>
217 <td><input_aria-label="Memory_Limit" type="text" name="memory_limit[]" class="sharif_input_tiny" value="<=?_$_problem['memory_limit']_?>"></td>
218 <td><input_aria-label="Allowed_Languages" type="text" name="languages[]" class="sharif_input_short2" value="<=?_$_problem['allowed_languages']_?>"></td>
219 <td><input_aria-label="Diff_Command" type="text" name="diff_cmd[]" class="sharif_input_tiny" value="<=?_$_problem['diff_cmd']_?>"></td>
220 <td><input_aria-label="Diff_Argument" type="text" name="diff_arg[]" class="sharif_input_tiny" value="<=?_$_problem['diff_arg']_?>"></td>
221 <td><input_aria-label="Upload_Only" type="checkbox" name="is_upload_only[]" class="check" value="<=?_$_problem['id']_?>"<?_$_problem['is_upload_only']_?>_checked'_&:_?></td>
222 <td><button_aria-label="Delete_Problem" type="button" class="sharif_input_tiny fa_fa-times-circle_fa-lg_color1_pointer"></i></button></td>
223 </tr>
224 <?php endforeach_?>
225 </tbody>
226 </table>
227 </p>
228 <div_class="shj_error"><=?_$_validationError->hasError('name.*')_?_$_validationError->getError('name.*')_&:_?></div>
229 <div_class="shj_error"><=?_$_validationError->hasError('score.*')_?_$_validationError->getError('score.*')_&:_?></div>
230 <div_class="shj_error"><=?_$_validationError->hasError('c_time_limit/*') ? $_validationError->getError('c_time_limit.*') : '' ?></div>

```

```

231 <div class="shj_error"><?= $validationError->hasError('python_time_limit.*') ? $validationError->getError('python_time_limit.*') : '' ?></div>
232 <div class="shj_error"><?= $validationError->hasError('java_time_limit.*') ? $validationError->getError('java_time_limit.*') : '' ?></div>
233 <div class="shj_error"><?= $validationError->hasError('memory_limit.*') ? $validationError->getError('memory_limit.*') : '' ?></div>
234 <div class="shj_error"><?= $validationError->hasError('languages.*') ? $validationError->getError('languages.*') : '' ?></div>
235 <div class="shj_error"><?= $validationError->hasError('diff_cmd.*') ? $validationError->getError('diff_cmd.*') : '' ?></div>
236 <div class="shj_error"><?= $validationError->hasError('diff_arg.*') ? $validationError->getError('diff_arg.*') : '' ?></div>
237 <p class="input_p">
238   <input type="submit" value="<?= $edit ? 'Edit' : 'Add' ?> Assignment" class="sharif_input"/>
239 </p>
240 </form>
241 <?php $this->endSection() ?>
```

Kode C.2: add_notification.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: add_notification.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?><?= $notif_edit ? 'fa-edit' : 'fa-plus' ?><?= $this->endSection() ?>
8 <?= $this->section('title') ?><?= $notif_edit ? 'Edit' : 'New' ?><?= $this->endSection() ?>
9 <?= $this->section('head_title') ?><?= $notif_edit ? 'Edit' : 'New' ?><?= $this->endSection() ?>
10
11 <?= $this->section('other_assets')?>
12 <script type='text/javascript' src="<?= _base_url('assets/tinymce/tinymce.min.js') ?>"></script>
13 <script>
14 $(document).ready(function(){
15   tinymce.init({
16     selector: 'textarea#notif_text',
17     toolbar_items_size: 'small',
18     relative_urls: false,
19     width: 700,
20     height: 300,
21     resize: false,
22     plugins: 'directionality_emoticons_textcolor_link_code',
23     toolbar1: "undo redo | styleselect | bold italic | alignleft aligncenter alignright alignjustify | bullist numlist outdent indent | ltr rtl",
24     toolbar2: "forecolor backcolor | emoticons | link unlink anchor image media code | removeformat"
25   });
26 });
27 </script>
28 <?= $this->endSection() ?>
29
30 <?= $this->section('main_content') ?>
31 <?= form_open('notifications/'.($notif_edit ? 'edit/'.$notif_edit['id'] : 'add')) ?>
32 <?php if ($notif_edit) ?>
33   <input type="hidden" name="id" value="<?= $notif_edit['id'] ?>"/>
34 <?php endif ?>
35 <p class="input_p">
36   <label for="form_title" class="tiny">Title:</label>
37   <input id="form_title" name="title" type="text" class="sharif_input" value="<?= $notif_edit ? $notif_edit['title'] : '' ?>"/>
38 </p>
39 <p class="input_p">
40   <label for="notif_text" class="tiny">Text:</label><br><br>
41   <textarea id="notif_text" name="text"><?= $notif_edit ? $notif_edit['text'] : '' ?></textarea>
42 </p>
43 <p class="input_p">
44   <input type="submit" value="<?= $notif_edit ? 'Save' : 'Add' ?>" class="sharif_input"/>
45 </p>
46 </form>
47 <?= $this->endSection() ?>
```

Kode C.3: add_user_result.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: add_user_result.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?php if($ok): ?>
7 <p class="shj_ok">These users added successfully:</p>
8 <ol>
9   <?php foreach ($ok as $item): ?>
10    <li>Username: <?= $item[0] ?> Email: <?= $item[1] ?> Display Name: <?= $item[2] ?> Password: <code><?= $item[3] ?></code> Role: <?= $item[4] ?> </li>
11  <?php endforeach ?>
12 </ol>
13 <?php endif ?>
14 <?php if ($error): ?>
15 <p class="shj_error">Error adding these users:</p>
16 <ol>
17   <?php foreach($error as $item): ?>
18    <li>Username: <?= $item[0] ?> Email: <?= $item[1] ?> Display Name: <?= $item[2] ?> Password: <code><?= $item[3] ?></code> Role: <?= $item[4] ?> (<?= $item[5] ?>)</li>
19  <?php endforeach ?>
20 </ol>
21 <?php endif ?>
```

Kode C.4: add_user.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: add_user.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-plus<?= $this->endSection()?>
8 <?= $this->section('title') ?>Add Users<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Add Users<?= $this->endSection() ?>
10
11 <?= $this->section('title_menu') ?>
12   <span class="title_menu_item"><a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/users.md#add-users" target="_blank"><i class="fa_fa-question-circle_color6"></i> Help</a></span>
13 <?= $this->endSection() ?>
14
15
16
17 <?= $this->section('other_assets') ?>
18 <script type='text/javascript' src=<?=base_url('assets/js/taboverride.min.js')?>></script>
19 <script>
20   $(document).ready(function(){
21     tabOverride.set(document.getElementsByName('textarea'));
22     $switch = false;
23     $("textarea").keyup(function (e) {
24       if (e.which==27){
25         tabOverride.set(document.getElementsByName('textarea'),$switch);
26         $switch = !$switch;
27       }
28     });
29   });
30 </script>
31 <script>
32   $(document).ready(function(){
33     $("#add_users_button").click(function(){
34       $("#loading").css('display','inline');
35       $.post(
36         '<?=site_url('users/add')?>',
37         {
38           send_mail: ($("#send_mail").is(":checked")?1:0),
39           delay: ($("#delay").val()),
40           new_users:$("#new_users").val(),
41           shj_csrf_token: shj.csrf_token
42         }
43       ).done(function(data) {
44         $("#main_content").html(data);
45       }
46     );
47   });
48 });
49 </script>
50 <?= $this->endSection() ?>
51
52
53 <?= $this->section('main_content') ?>
54 <p>You can use this field to add multiple users at the same time.</p>
55 <ul>
56   <li>Usernames may contain lowercase letters or numbers and must be between 3 and 20 characters in length.</li>
57   <li>Passwords must be between 6 and 30 characters in length.</li>
58   <li>If you want to send passwords by email, do not add too many users at one time. This may result in mail delivery fail.</li>
59 </ul>
60 <p class="input_p">
61   <input type="checkbox" name="send_mail" id="send_mail" aria-label="Send_email"/> Send usernames and passwords by email (Waits <
62     input type="text" name="delay" id="delay" class="sharif_input_tiny" value="2" aria-label="Delay"/> second(s) before
63     sending each email, so please be patient).
64 </p>
65 <p class="input_p">
66   <span class="form_comment_clear">Press "esc" to enable/disable tabindent</span><br>
67 <textarea name="new_users" id="new_users" rows="20" cols="80" class="sharif_input" aria-label="Command_for_creating_new_user">
68 # Lines starting with a # sign are comments.
69 # Each line (except comments) represents a user.
70 # The syntax of each line is:
71 #
72 # USERNAME,EMAIL,DISPLAY-NAME,PASSWORD,ROLE
73 #
74 # Roles: admin head_instructor instructor student
75 # You can use RANDOM[n] for password to generate random n-digit password.
76 </p>
77 <input type="submit" class="sharif_input" id="add_users_button" value="Add_Users"/>
78 <span id="loading" style="display:none;"> Adding users... Please wait</span>
79 <?= $this->endSection() ?>

```

Kode C.5: delete_assignment.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: delete_assignment.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-times<?= $this->endSection()?>
8 <?= $this->section('title') ?>Delete Assignment<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Delete Assignment<?= $this->endSection() ?>
10
11 <?= $this->section('main_content') ?>
12 <p>Are you sure you want to delete this assignment?</p>
13 <p>

```

```

14     Assignment id: <?= $id ?><br>
15     Assignment name: <span dir="auto"><?= $name ?></span>
16 </p>
17 <p>All test cases, submission results and submitted files will be deleted.</p>
18 <p>You may want to keep a backup of this assignment before deleting.</p>
19 <?= form_open("assignments/delete/$id") ?>
20 <input type="hidden" name="delete" value="delete"/>
21 <p class="input_p">
22   <input type="submit" class="btn_shj-red" value="Delete_this_assignment"/>
23   <a href="= site_url('assignments') ?&gt;" class="btn_shj-blue"&gt;Do not delete&lt;/a&gt;
24 &lt;/p&gt;
25 &lt;/form&gt;
26 &lt;?= $this-&gt;endSection() ?&gt;
</pre

```

Kode C.6: edit_problem_html.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: edit_problem_html.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-edit<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Edit Problem Description<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Edit Problem Description<?= $this->endSection() ?>
10
11
12
13 <?= $this->section('other_assets') ?>
14 <script type='text/javascript' src="= base_url('assets/tinymce/tinymce.min.js') ?&gt;"&gt;&lt;/script&gt;
15 &lt;script&gt;
16 $(document).ready(function(){
17   tinymce.init({
18     selector: 'textare',
19     toolbar_items_size: 'small',
20     relative_urls: false,
21     width: 700,
22     height: 300,
23     resize: false,
24     plugins: 'directionality_emoticons_textcolor_link_code',
25     toolbar1: 'undo_redo|styleselect|bold_italic|alignleft_aligncenter_alignright_alignjustify|bullist_numlist_outdent
26       indent_ltr rtl',
27     toolbar2: 'forecolor_backcolor|emoticons|link_unlink_anchor_image_media_code|removeformat'
28   });
29 });
30 &lt;/script&gt;
31 &lt;?= $this-&gt;endSection() ?&gt;
32
33
34 &lt;?= $this-&gt;section('main_content') ?&gt;
35 &lt;p&gt;
36   Assignment &lt;?= $description_assignment['id'] ?&gt; (&lt;span dir="auto"&gt;&lt;?= $description_assignment['name'] ?&gt;&lt;/span&gt;)&lt;br&gt;
37   Problem &lt;?= $problem['id'] ?&gt;
38 &lt;/p&gt;
39 &lt;p&gt;
40   &lt;i class="fa fa-warning_color3"&gt;&lt;/i&gt;
41   When you edit as html, the markdown code will be removed.
42 &lt;/p&gt;
43 &lt;?= form_open("problems/edit/html/".$description_assignment['id'].'/'.$problem['id']) ?&gt;
44 &lt;p class="input_p"&gt;
45   &lt;textarea name="text"&gt;&lt;?= $problem['description'] ?&gt;&lt;/textarea&gt;
46 &lt;/p&gt;
47 &lt;p class="input_p"&gt;
48   &lt;input type="submit" value="Save" class="sharif_input"/&gt;
49 &lt;/p&gt;
50 &lt;/form&gt;
51 &lt;?= $this-&gt;endSection() ?&gt;
</pre

```

Kode C.7: edit_problem_md.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: edit_problem_md.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-edit<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Edit Problem Description (Markdown)<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Edit Problem Description (Markdown)<?= $this->endSection() ?>
10
11
12
13 <?= $this->section('other_assets') ?>
14 <script type='text/javascript' src="= base_url('assets/js/taboverride.min.js') ?&gt;"&gt;&lt;/script&gt;
15 &lt;script&gt;
16 $(document).ready(function(){
17   tabOverride.set(document.getElementById('md_editor'));
18   $switch = false;
19   $("textare").keyup(function (e) {
20     if (e.which==27){
21       tabOverride.set(document.getElementById('md_editor'),$switch);
22       $switch = !$switch;
23     }
24   });
</pre

```

```

25  });
26 </script>
27 <?= $this->endSection() ?>
28
29
30
31 <?= $this->section('main_content') ?>
32 <div class="pull-right_md_cheatsheet">
33 <h2>Markdown Cheatsheet</h2>
34 From <a href="http://daringfireball.net/projects/markdown/dingus">Daring Fireball</a>
35 <h3>Headers</h3>
36
37 <p>Setext-style:</p>
38
39 <pre><code>Header 1
40 =====
41 Header 2
42 -----
43 </code></pre>
44
45 <p>atx-style (closing #'s are optional):</p>
46
47 <pre><code># Header 1 #
48
49 ## Header 2 ##
50
51 ##### Header 6
52 </code></pre>
53
54 <h3>Phrase Emphasis</h3>
55
56 <pre><code>*italic* **bold**
57 _italic_ __bold__
58 </code></pre>
59
60 <h3>Code Spans</h3>
61
62 <pre><code>'&lt;code&gt;' spans are delimited
63 by backticks.</code></pre>
64
65 <h3>Fenced Code Blocks</h3>
66
67 <pre>
68 '''
69 function test() {
70     printf("Hello_World!\n");
71 }
72 '''</pre>
73 <h3>Syntax Highlighting</h3>
74 <code>c</code>, <code>cpp</code>, <code>python</code> and <code>java</code> supported.
75 <pre>
76 '''python
77 def test:
78     print ("Python_Syntax")
79 '''</pre>
80
81 <h3>Preformatted Code Blocks</h3>
82
83 <p>Indent every line of a code block by at least 4 spaces or 1 tab.</p>
84
85 <pre><code>This is a normal paragraph.
86
87     This is a preformatted
88     code block.
89 </code></pre>
90
91 <h3>Lists</h3>
92
93 <p>Ordered, without paragraphs:</p>
94
95 <pre><code>1. Foo
96 2. Bar
97 </code></pre>
98
99 <p>Unordered, with paragraphs:</p>
100
101 <pre><code>* A list item.
102
103     With multiple paragraphs.
104
105     * Bar
106 </code></pre>
107
108 <p>You can nest them:</p>
109
110 <pre><code>* Item 1
111     * item
112     * Item 2
113         1. item 1
114         2. item 2
115             * item
116             3. item 3
117     * Item 3
118 </code></pre>
119
120 <h3>Links</h3>
121
122 <p>Inline:</p>
123

```

```

124| <pre><code>An [example](http://url.com/ "Title")
125| </code></pre>
126|
127| <p>Reference-style labels (titles are optional):</p>
128|
129| <pre><code>An [example][id]. Then, anywhere
130| else in the doc, define the link:
131|
132|     [id]: http://example.com/ "Title"
133| </code></pre>
134| <h3>Images</h3>
135|
136| <p>Inline (titles are optional):</p>
137|
138| <pre><code>! [alt text] (/path/img.jpg "Title")
139| </code></pre>
140|
141| <p>Reference-style:</p>
142|
143| <pre><code>! [alt text][id]
144|
145|     [id]: /url/to/img.jpg "Title"
146| </code></pre>
147| <h3>Blockquotes</h3>
148|
149| <pre><code>&gt; Email-style angle brackets
150| &gt; are used for blockquotes.
151|
152| &gt; &gt; And, they can be nested.
153|
154| &gt; ##### Headers in blockquotes
155| &gt;
156| &gt; * You can quote a list.
157| &gt; * Etc.
158| </code></pre>
159| <h3>Horizontal Rules</h3>
160|
161| <p>Three or more dashes or asterisks:</p>
162|
163| <pre><code>---
164|
165| * * *
166|
167| - - -
168| </code></pre>
169|
170| <h3>Manual Line Breaks</h3>
171|
172| <p>End a line with two or more spaces:</p>
173|
174| <pre><code>Roses are red,
175| Violets are blue.
176| </code></pre>
177| </div>
178|
179| <div id="md_div">
180|   <p>
181|     Assignment <?= $description_assignment['id'] ?> (<span dir="auto"><?= $description_assignment['name'] ?></span>)<br>
182|     Problem <?= $problem['id'] ?>
183|   </p>
184|   <?= form_open("problems/edit/md/" . $description_assignment['id'] . '/' . $problem['id']) ?>
185|   <span class="form_comment_clear">Press "esc" to enable/disable tabindent</span><br>
186|   <p class="input_p">
187|     <textarea dir="auto" name="text" rows="30" cols="75" class="sharif_input" id="md_editor" aria-label="Markdown_Editor"><?= $problem['description'] ?></textarea>
188|   </p>
189|   <p class="input_p">
190|     <input type="submit" value="Save" class="sharif_input"/>
191|   </p>
192| </form>
193| </div>
194|
195| <?= $this->endSection() ?>

```

Kode C.8: edit_problem_plain.php

```

1| <!-- {#
2| # Sharif Judge
3| # file: edit_problem_plain.twig
4| # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5| #} -->
6|
7| <?= $this->extend('templates/base') ?>
8| <?= $this->section('icon') ?>fa-edit<?= $this->endSection() ?>
9| <?= $this->section('title') ?>Edit Problem Description (Plain HTML)<?= $this->endSection() ?>
10| <?= $this->section('head_title') ?>Edit Problem Description (Plain HTML)<?= $this->endSection() ?>
11|
12|
13|
14| <?= $this->section('other_assets') ?>
15| <script type='text/javascript' src="= _base_url('assets/js/taboverride.min.js') ?&gt;"&gt;&lt;/script&gt;
16| &lt;script&gt;
17|   $(document).ready(function(){
18|     tabOverride.set(document.getElementById('md_editor'));
19|   });
20| &lt;/script&gt;
21| &lt;?= $this-&gt;endSection() ?&gt;
22|
</pre

```

```

23
24  <?= $this->section('main_content') ?>
25  <p>
26      Assignment <?= $description_assignment['id'] ?> (<span dir="auto"><?= $description_assignment['name'] ?></span>)<br>
27      Problem <?= $problem['id'] ?>
28  </p>
29  <p>
30      <i class="fa fa-warning_color3"></i>
31      When you edit as html, the markdown code will be removed.
32  </p>
33  <?= form_open("problems/edit/html/".$description_assignment['id'].'/'.$problem['id']) ?>
34  <p class="input_p">
35      <textarea name="text" rows="30" cols="80" class="sharif_input" id="html_editor" aria-label="HTML_Editor"><?= $problem['
36          description'] ?></textarea>
37  </p>
38  <p class="input_p">
39      <input type="submit" value="Save" class="sharif_input"/>
40  </p>
41  </form>
42  <?= $this->endSection() ?>

```

Kode C.9: install.php

```

1  <!-- {#
2  # SharIF Judge
3  # file: install.twig
4  # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5  #} -->
6  <?= $this->include('templates/simple_header')?>
7  <?php if($installed): ?>
8      <div id="install_success">
9          <h2>SharIF Judge Installed</h2>
10         <?php if(!$key_changed): ?>
11             <p class="shj_error">
12                 <code>app/Config/Encryption.php</code> is not writable by PHP.
13             </p>
14             <p>
15                 For security reasons, you should change the encryption key manually.<br>
16                 Open <code>app/Config/Encryption.php</code> and change the encryption key in this line:
17             </p>
18             <pre>$encryption_key = '<?= $enc_key ?>';</pre>
19             <p>
20                 The key should be a 32-characters string as random as possible, with numbers and uppercase and lowercase letters.<
21                 br>
22                 You can use this random string: <code><?= $random_key ?></code>
23             </p>
24             <br>
25         <?php endif ?>
26         <p class="shj_ok">SharIF Judge installed! Now you can <a href="<?= $base_url('login') ?>">login</a>. </p>
27     </div>
28
29 <?php else: ?>
30     <div id="install_form">
31         <h2>SharIF Judge Installation</h2>
32         <?= form_open('install') ?>
33         <p class="input_p">
34             <label for="form_username">Admin username:</label>
35             <input id="form_username" class="sharif_input" type="text" name="username" required="required" pattern="[0-9A-Za-z
36                 ]{3,20}" title="The Username field must be between 3 and 20 characters in length, and contain only alpha-numeric
37                 characters" value="<?= $set_value('username') ?>" autofocus>
38             <div class="shj_error"><?= $validationError->hasError('username') ? $validationError->getError('username') : '' ?><
39                 div>
40             </p>
41             <p class="input_p">
42                 <label for="form_email">Admin email:</label>
43                 <input id="form_email" type="email" autocomplete="off" name="email" required="required" class="sharif_input" value=
44                     "<?= $set_value('email') ?>"/>
45                 <div class="shj_error"><?= $validationError->hasError('email') ? $validationError->getError('email') : '' ?></div>
46             </p>
47             <p class="input_p">
48                 <label for="form_password">Admin password:</label>
49                 <input id="form_password" type="password" name="password" required="required" pattern="[0-9A-Za-z]{6,20}" title="The_
50                     Password field must be between 6 and 30 characters in length, and contain only alpha-numeric characters" class="sharif_input"/>
51                 <div class="shj_error"><?= $validationError->hasError('password') ? $validationError->getError('password') : '' ?><
52                     div>
53             </p>
54             <p class="input_p">
55                 <label for="form_password_2">Password, again:</label>
56                 <input id="form_password_2" type="password" name="password_again" required="required" class="sharif_input"/>
57                 <div class="shj_error"><?= $validationError->hasError('password_again') ? $validationError->getError('password_again')
58                     : '' ?></div>
59             </p>
60         <?= form_close() ?>
61     </div>
62 <?php endif ?>
63 </body>
64 </html>

```

Kode C.10: logs.php

```

1| <!-- {#
2| # SharIF Judge
3| # file: logs.twig
4| # author: Stillmen Vallian <stillmen.v@gmail.com>
5| #} -->
6| <?= $this->extend('templates/base') ?>
7| <?= $this->section('icon') ?>fa-book<?= $this->endSection()?>
8| <?= $this->section('title') ?>24-hour Log<?= $this->endSection() ?>
9| <?= $this->section('head_title') ?>24-hour Log<?= $this->endSection() ?>
10|
11| <?= $this->section('main_content') ?>
12|
13| <p style="text-align:justify">Use this table to detect account lendings between students in a pre-seated exam environment. In a pre-seated environment, student should have logged in from only one IP address within the exam time period. Whenever a user logs in from different IP address in the past 24 hours, last column of this table will show link to the previous login that happened. In such case, proctor can analyse from those two logins, whether the same login has been used in different devices (meaning the account had been lent).</p>
14|
15| <table class="sharif_table">
16|   <thead>
17|     <tr>
18|       <th>#</th>
19|       <th>Login ID</th>
20|       <th>Username</th>
21|       <th>IP Address</th>
22|       <th>Login Time</th>
23|       <th>Log from different IP (< 24 hours)</th>
24|     </tr>
25|   </thead>
26|   <?php foreach ($logs as $index => $log): ?>
27|     <tr>
28|       <td><?= $index + 1 ?></td>
29|       <td id="<?=$log['login_id']?>"><?= $log['login_id'] ?></td>
30|       <td><?= $log['username'] ?></td>
31|       <td><?= $log['ip_address'] ?></td>
32|       <td><?= $log['timestamp'] ?></td>
33|       <td><a href="#<?=$log['last_24h_login_id']?>"><?= $log['last_24h_login_id'] ?></a></td>
34|     </tr>
35|   </php endforeach ?>
36| </table>
37| <?= $this->endSection() ?>

```

Kode C.11: moss.php

```

1| <!-- {#
2| # SharIF Judge
3| # file: moss.twig
4| # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5| #} -->
6| <?= $this->extend('templates/base') ?>
7| <?= $this->section('icon') ?>fa-shield<?= $this->endSection()?>
8| <?= $this->section('title') ?>Detect Similar Codes<?= $this->endSection() ?>
9| <?= $this->section('head_title') ?>Detect Similar Codes<?= $this->endSection() ?>
10|
11|
12|
13| <?= $this->section('other_assets') ?>
14|   <script type='text/javascript' src="<?=base_url('assets/js/taboverride.min.js')?>"></script>
15|   <script>
16|     $(document).ready(function(){
17|       tabOverride.set(document.getElementById('md_editor'));
18|     });
19|   </script>
20| <?= $this->endSection() ?>
21|
22|
23|
24|
25| <?= $this->section('main_content') ?>
26| <h3>What is Moss?</h3>
27| <p>
28|   <a href="http://theory.stanford.edu/~aiken/moss/" target="_blank">Moss</a> (for a Measure Of Software Similarity)
29|   is an automatic system for determining the similarity of programs.
30|   To date, the main application of Moss has been in detecting plagiarism in programming classes. Since its
31|   development in 1994, Moss has been very effective in this role. The algorithm behind moss is a significant
32|   improvement over other cheating detection algorithms.
33| </p>
34|
35| <br>
36|
37| <h3>Moss user id</h3>
38| <?php if (!$moss_userid): ?>
39|   <p class="shj_error">You have not entered your Moss user id.</p>
40| <?php endif ?>
41| <p>
42|   Read <a href="http://theory.stanford.edu/~aiken/moss/" target="_blank">this page</a> and register for Moss,
43|   then find your user id in the script sent to your email by Moss and enter your user id here.
44| </p>
45| <?= form_open('moss/update/'.$moss_assignment['id']) ?>
46| <p class="input_p">
47|   <label for="moss_uid">Your Moss user id is:</label>
48|   <input id="moss_uid" type="text" name="moss_userid" class="sharif_input" value="<?=$moss_userid?>"/>
49| </p>
50| <input type="submit" class="sharif_input" value="Save"/>
51| </form>
52|
53| <br>
54|

```

```

55| <h3>Detect similar submissions of assignment "<span dir="auto"><?=_$moss_assignment['name']_?></span>" :</h3>
56| <p>
57| <?= form_open('moss/'.$moss_assignment['id']) ?>
58| <input type="hidden" name="detect" value="detect" />
59| You can send final submissions of assignment "<span dir="auto"><?=_$moss_assignment['name']_?></span>" to Moss
60| by clicking on this button.<br>
61| Zip and PDF files will not be sent.<br>
62| It may take a minute. Please be patient.<br>
63| <input type="submit" class="sharif_input" value="Detect_similar_codes"/>
64| </form>
65| </p>
66|
67| <br>
68|
69| <h3>Moss results for assignment "<span dir="auto"><?=_$moss_assignment['name']_?></span>" :</h3>
70| <p>
71|     Links will expire after some time. (last update: <?= $update_time ?>) <br>
72|     <ul>
73|     <?php foreach ($moss_problems as $i => $moss_problem): ?>
74|         <li>Problem <?= $i ?:>
75|             <?php if ($moss_problem == null): ?>
76|                 Link Not Found.
77|             <?php elseif (!$moss_problem): ?>
78|                 Link Not Found. Maybe you have entered wrong user id.
79|             <?php else: ?>
80|                 <a href=<?=_$moss_problem_?>" target="_black"><?= $moss_problem ?></a>
81|             <?php endif ?>
82|         </li>
83|     <?php endforeach ?>
84|     </ul>
85| </p>
86| <?= $this->endSection() ?>

```

Kode C.12: queue.php

```

1<!-- {#
2# SharIF Judge
3# file: queue.twig
4# author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5#} -->
6<?= $this->extend('templates/base') ?>
7<?= $this->section('icon') ?>fa-play<?= $this->endSection() ?>
8<?= $this->section('title') ?>Submission Queue<?= $this->endSection() ?>
9<?= $this->section('head_title') ?>Submission Queue<?= $this->endSection() ?>
10
11
12
13<?= $this->section('other_assets') ?>
14<script>
15    $(document).ready(function(){
16        $(".shj_act").click(function(){
17            var action=$(this).attr('id');
18            $.post(
19                '<?=site_url('queue')_?>/' + action,
20                {shj_csrf_token: shj.csrf_token},
21                function(data){
22                    if (data=='success')
23                        location.reload();
24                }
25            );
26        });
27    });
28</script>
29<?= $this->endSection() ?>
30
31
32
33<?= $this->section('main_content') ?>
34<p>
35    <?php if ($working): ?>
36        <i class="fa fa-play fa-lg color6"></i> Queue is working
37    <?php else: ?>
38        <i class="fa fa-pause fa-lg color10"></i> Queue is not working
39    <?php endif ?>
40    | Total submissions in queue: <?= count($queue) ?>
41</p>
42<p>
43    <a href="#" class="shj_act" id="pause"><i class="fa fa-pause"></i> Pause</a> |
44    <a href="#" class="shj_act" id="resume"><i class="fa fa-play"></i> Resume</a> |
45    <a href="#" class="shj_act" id="empty_queue"><i class="fa fa-times-circle"></i> Empty Queue</a>
46</p>
47<table class="sharif_table">
48    <thead>
49        <tr>
50            <th>#</th>
51            <th>Submit ID</th>
52            <th>Username</th>
53            <th>Assignment</th>
54            <th>Problem</th>
55            <th>Type (judge/rejudge)</th>
56        </tr>
57    </thead>
58    <?php foreach ($queue as $index => $item): ?>
59        <tr>
60            <td><?= $index+1 ?></td>
61            <td><?= $item['submit_id'] ?></td>
62            <td><?= $item['username'] ?></td>
63            <td><?= $item['assignment'] ?> (<span dir="auto"><?= $all_assignments[$item['assignment']] ['name'] ?></span>)</td>

```

```

64      <td><?= $item['problem'] ?></td>
65      <td><?= $item['type'] ?></td>
66    </tr>
67  </php endforeach ?>
68 </table>
69 <?= $this->endSection() ?>
```

Kode C.13: rejudge.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: rejudge.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-refresh<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Rejudge<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Rejudge<?= $this->endSection() ?>
10
11
12 <?= $this->section('main_content') ?>
13 <p>
14   Selected Assignment: <span dir="auto"><?= $user->selected_assignment['name'] ?></span>
15 </p>
16 <p>
17   By clicking on rejudge, all submissions of selected problem will change to <code>PENDING</code> state. Then
18   SharIF Judge rejudges them one by one.
19 </p>
20 <p>
21   If you want to rejudge a single submission, you can click on rejudge button in <a href="= site_url('submissions/all') ?&gt;"&gt;
22     All Submissions&lt;/a&gt; or &lt;a href="<?= site_url('submissions/final') ?&gt;"&gt;Final Submissions&lt;/a&gt; page.
23 &lt;/p&gt;
24 &lt;?php foreach ($problems as $problem): ?&gt;
25   &lt;form_open('rejudge') ?&gt;
26     &lt;input type="hidden" name="problem_id" value="<?= $problem['id'] ?&gt;"/&gt;
27     &lt;input type="submit" class="sharif_input" value="Rejudge_Problem_&lt;?= $problem['id'] ?&gt;_(&lt;?= $problem['name'] ?&gt;)"/&gt;
28   &lt;/form&gt;
29 &lt;?php endforeach ?&gt;
30 &lt;?php foreach($msg as $message): ?&gt;
31   &lt;p class="shj_ok"&gt;&lt;?= $message ?&gt;&lt;/p&gt;
32 &lt;?php endforeach ?&gt;
33 &lt;?= $this-&gt;endSection() ?&gt;</pre

```

Kode C.14: settings.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: settings.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-gear<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Settings<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Settings<?= $this->endSection() ?>
10
11 <script type='text/javascript' src="= base_url('assets/js/taboverride.min.js') ?&gt;"/&gt;&lt;/script&gt;
12 &lt;script&gt;
13 $(document).ready(function(){
14   tabOverride.set(document.getElementsByTagName('textare'));
15   $switch = false;
16   $("textare").keyup(function (e) {
17     if (e.which==27){
18       tabOverride.set(document.getElementsByTagName('textare'),$switch);
19       $switch = !$switch;
20     }
21   });
22 });
23 &lt;/script&gt;
24
25 &lt;span class="title_menu_item"&gt;
26   &lt;a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/settings.md" target="_blank"&gt;&lt;i class="fa fa-question-circle color6"&gt;&lt;/i&gt; Help&lt;/a&gt;
27 &lt;/span&gt;
28
29 &lt;?= $this-&gt;section('main_content') ?&gt;
30 &lt;p class="input_p"&gt;
31 &lt;?php if ($form_status == 'ok'): ?&gt;
32   &lt;div class="shj_ok"&gt;Settings updated successfully.&lt;/div&gt;
33 &lt;?php elseif ($form_status == 'error'): ?&gt;
34   &lt;div class="shj_error"&gt;Error updating settings.&lt;/div&gt;
35 &lt;?php endif ?&gt;
36 &lt;?php foreach ($errors as $error) :?&gt;
37   &lt;div class="shj_error"&gt;&lt;?= $error ?&gt;&lt;/div&gt;
38 &lt;?php endforeach ?&gt;
39 &lt;?php if (!($defc)): ?&gt;
40   &lt;div class="shj_error"&gt;"Tester_path" is not correct.&lt;/div&gt;
41 &lt;?php endif ?&gt;
42 &lt;/p&gt;
43 &lt;?= form_open('settings/update') ?&gt;
44   &lt;p class="input_p"&gt;
45     &lt;label for="form_timezone"&gt;
46       Timezone&lt;br&gt;
47       &lt;a target="_blank" href="http://www.php.net/manual/en/timezones.php"&gt;list of timezones&lt;/a&gt;
48     &lt;/label&gt;</pre

```

```

49|     <span class="form_comment_timer"></span><br>
50|     <input id="form_timezone" type="text" name="timezone" class="sharif_input_medium" value=<?=$_timezone_?>/>
51|     <div class="shj_error"><?= $validationError->hasError('timezone') ? $validationError->getError('timezone') : '' ?></div>
52|   </p>
53|   <p class="input_p">
54|     <label for="form_week">Week Start Day</label>
55|     <select id="form_week" name="week_start" class="sharif_input">
56|       <option value="0" <?= ($week_start == 0) ? 'selected="selected"' : '' ?>>Sunday</option>
57|       <option value="1" <?= ($week_start == 1) ? 'selected="selected"' : '' ?>>Monday</option>
58|       <option value="2" <?= ($week_start == 2) ? 'selected="selected"' : '' ?>>Tuesday</option>
59|       <option value="3" <?= ($week_start == 3) ? 'selected="selected"' : '' ?>>Wednesday</option>
60|       <option value="4" <?= ($week_start == 4) ? 'selected="selected"' : '' ?>>Thursday</option>
61|       <option value="5" <?= ($week_start == 5) ? 'selected="selected"' : '' ?>>Friday</option>
62|       <option value="6" <?= ($week_start == 6) ? 'selected="selected"' : '' ?>>Saturday</option>
63|     </select>
64|   </p>
65|   <p class="input_p">
66|     <label for="form_t_path">Full Path to <code>tester</code></label>
67|     <input id="form_t_path" type="text" name="tester_path" class="sharif_input_medium" value=<?=$_tester_path_?>/>
68|   </p>
69|   <p class="input_p">
70|     <label for="form_a_path">Full Path to <code>assignments</code></label>
71|     <input id="form_a_path" type="text" name="assignments_root" class="sharif_input_medium" value=<?=$_assignments_root_?>/>
72|   </p>
73|   <p class="input_p">
74|     <label for="form_up_limit">Upload Size Limit (kB)</label>
75|     <input id="form_up_limit" type="text" name="file_size_limit" class="sharif_input_medium" value=<?=$_file_size_limit_?>/>
76|     <div class="shj_error"><?= $validationError->hasError('file_size_limit') ? $validationError->getError('file_size_limit') : '' ?></div>
77|   </p>
78|   <p class="input_p">
79|     <label for="form_out_limit">Output Size Limit (kB)</label>
80|     <input id="form_out_limit" type="text" name="output_size_limit" class="sharif_input_medium" value=<?=$_output_size_limit_?>><br>
81|     <span class="form_comment_clear">Sets a limit for size of output file generated by submitted code</span>
82|     <div class="shj_error"><?= $validationError->hasError('output_size_limit') ? $validationError->getError('output_size_limit') : '' ?></div>
83|   </p>
84|   <p class="input_p">
85|     <label for="form_rpp_all">Results Per Page</label>
86|     <input id="form_rpp_all" type="text" name="rpp_all" class="sharif_input_medium" value=<?=$_results_per_page_all_?>/>
87|     <p class="form_comment_clear">In 'All_Submissions'<br>Enter 0 for no limit</p>
88|     <div class="shj_error"><?= $validationError->hasError('rpp_all') ? $validationError->getError('rpp_all') : '' ?></div>
89|   </p>
90|   <p class="input_p">
91|     <label for="form_rpp_final">Results Per Page</label>
92|     <input id="form_rpp_final" type="text" name="rpp_final" class="sharif_input_medium" value=<?=$_results_per_page_final_?>>
93|     <p class="form_comment_clear">In 'Final_Submissions'<br>Enter 0 for no limit</p>
94|     <div class="shj_error"><?= $validationError->hasError('rpp_final') ? $validationError->getError('rpp_final') : '' ?></div>
95|   </p>
96|   <p class="input_p">
97|     <input id="form_en_reg" type="checkbox" name="enable_registration" value="1" <?= $enable_registration ? 'checked' : '' ?>/>
98|     <label for="form_en_reg">Registration</label><br>
99|     <span class="form_comment">Open Public Registration.</span>
100|   </p>
101|   <p class="input_p">
102|     <label for="form_reg_code">Registration Code</label>
103|     <input id="form_reg_code" type="text" name="registration_code" class="sharif_input_medium" value=<?=$_registration_code_?>><br>
104|     <p class="form_comment_clear">If you want to enable registration (above option), It is better to give a registration code<br>
105|       to students in your class for validating registration. Enter 0 to disable.</p>
106|     <div class="shj_error"><?= $validationError->hasError('registration_code') ? $validationError->getError('registration_code') : '' ?></div>
107|   </p>
108|   <p class="input_p">
109|     <input id="form_log" type="checkbox" name="enable_log" value="1" <?= $enable_log ? 'checked' : '' ?>/>
110|     <label for="form_log">Log</label><br>
111|     <span class="form_comment">Enable Log</span>
112|   </p>
113|   <p class="input_p">
114|     <input id="form_lock_student_display_name" type="checkbox" name="lock_student_display_name" value="1" <?= $lock_student_display_name ? 'checked' : '' ?>/>
115|     <label for="form_lock_student_display_name">Lock Student's Display Name</label><br>
116|     <span class="form_comment">Student's can't change their display name</span>
117|   </p>
118|   <p class="input_p">
119|     <label for="form_late_rule">Default_Coefficient_Rule</label>
120|     <span class="form_comment_clear">PHP_script_without_&lt;?php_?&gt;;_tags</span><br>
121|     <span class="form_comment_clear">Press esc to enable/disable tab indent</span><br>
122|     <textarea id="form_late_rule" name="default_late_rule" rows="15" class="sharif_input_add_text_clear"><?=$_default_late_rule_?></textarea>
123|   </p>
124|   <h2 class="shj_form">Email</h2>
125|   <p class="input_p">
126|     <label for="form_mail_from">Send Emails From</label>
127|     <input id="form_mail_from" type="text" name="mail_from" class="sharif_input_medium" value=<?=$_mail_from_?>/>
128|     <div class="shj_error"><?= $validationError->hasError('mail_from') ? $validationError->getError('mail_from') : '' ?></div>
129|   </p>
130|   <p class="input_p">
131|     <label for="form_mail_name">Send Emails From Name</label>
132|     <input id="form_mail_name" type="text" name="mail_from_name" class="sharif_input_medium" value=<?=$_mail_from_name_?>/>
133|     <div class="shj_error"><?= $validationError->hasError('mail_from_name') ? $validationError->getError('mail_from_name') : '' ?></div>
134|   </p>
135|   <p></p>
136| 
```

```

137 137< p class="input_p">
138 138    <label_for="form_mail_reset">Password_Reset_Email</label>
139 139    <span class="form_comment">You_can_use_{SITE_URL},{RESET_LINK}_and_{VALID_TIME}</span><br>
140 140    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
141 141    <textarea_id="form_mail_reset"_name="reset_password_mail"_rows="15"_class="sharif_input_add_text_clear"><?=$_
142 142      $reset_password_mail_?></textarea>
143 143  </p>
144 144  <label_for="form_mail_add">Add_User_Email</label>
145 145    <span class="form_comment_clear">You_can_use_{SITE_URL},{LOGIN_URL},{ROLE},{USERNAME}_and_{PASSWORD}</span><br>
146 146    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
147 147    <textarea_id="form_mail_add"_name="add_user_email"_rows="15"_class="sharif_input_add_text_clear"><?=$_$add_user_email_?><
148 148      textarea>
149 149  </p>
150 150
151 151  <h2 class="shj_form">
152 152    <Sandboxing><span class="title_menu_item">
153 153      <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md" target="_blank"><i class="fa fa-
154 154        question-circle_color11"></i> Help</a>
155 155    </span>
156 156  </h2>
157 157  <p class="input_p">
158 158    <input_id="form_easysandbox"_type="checkbox"_name="enable_easysandbox"_value="1"><?=$_enable_easysandbox_?>'checked'>< />
159 159  </p>
160 160  <label_for="form_easysandbox">EasySandbox</label>
161 161  <p class="form_comment">Enable_EasySandbox_for_C/C++.<br>
162 162    You_must<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md#build-easysandbox" target="_blank">
163 163      build_EasySandbox</a>_before_enabling_it.<br>
164 164      <?php_if_!$sandbox_built:>?>
165 165      <span style="color:red;">You_have_not_built_EasySandbox_yet.</span>
166 166    <?php_endif_?>
167 167  </p>
168 168  <p class="input_p">
169 169    <input_id="form_java_policy"_type="checkbox"_name="enable_java_policy"_value="1"><?=$_enable_java_policy_?>'checked'>< />
170 170  <label_for="form_java_policy">Java_Policy</label>
171 171  </p>
172 172  <p class="form_comment">
173 173    Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/sandboxing.md#java-sandboxing" target="_blank">Java
174 174      Sandboxing</a>
175 175  </p>
176 176
177 177  <h2 class="shj_form">
178 178    <Shield><span class="title_menu_item">
179 179      <a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_blank"><i class="fa fa-question-
180 180        circle_color11"></i> Help</a>
181 181    </span>
182 182  </h2>
183 183
184 184  <p class="input_p">
185 185    <input_id="form_c_sh"_type="checkbox"_name="enable_c_shield"_value="1"><?=$_enable_c_shield_?>'checked'>< /><label_
186 186      for="form_c_sh">C_Shield</label><br>
187 187    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_
188 188      _blank">Shield</a>_for_C</span>
189 189  <p class="input_p">
190 190    <input_id="form_cpp_sh"_type="checkbox"_name="enable_cpp_shield"_value="1"><?=$_enable_cpp_shield_?>'checked'>< />
191 191    <label_for="form_cpp_sh">C++_Shield</label><br>
192 192    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_
193 193      _blank">Shield</a>_for_C++</span>
194 194  <p class="input_p">
195 195    <input_id="form_py2_sh"_type="checkbox"_name="enable_py2_shield"_value="1"><?=$_enable_py2_shield_?>'checked'>< />
196 196    <label_for="form_py2_sh">Python_2_Shield</label><br>
197 197    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_
198 198      _blank">Shield</a>_for_Python_2</span>
199 199  <p class="input_p">
200 200    <input_id="form_py3_sh"_type="checkbox"_name="enable_py3_shield"_value="1"><?=$_enable_py3_shield_?>'checked'>< />
201 201    <label_for="form_py3_sh">Python_3_Shield</label><br>
202 202    <span class="form_comment">Enable<a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/shield.md" target="_
203 203      _blank">Shield</a>_for_Python_3</span>
204 204  <p class="input_p">
205 205    <label_for="form_def_c">Shield_Rules_(for_C)</label>
206 206    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
207 207    <textarea_id="form_def_c"_name="def_c"_rows="15"_class="sharif_input_add_text_clear"><?=$_$defc_?></textarea>
208 208  </p>
209 209  <label_for="form_def_cpp">Shield_Rules_(for_C++)</label>
210 210  <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
211 211  <textarea_id="form_def_cpp"_name="def_cpp"_rows="15"_class="sharif_input_add_text_clear"><?=$_$defcpp_?></textarea>
212 212  </p>
213 213  <p class="input_p">
214 214    <label_for="form_shield_py2">Shield_(for_Python_2)</label>
215 215    <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
216 216    <textarea_id="form_shield_py2"_name="shield_py2"_rows="15"_class="sharif_input_add_text_clear"><?=$_$shield_py2_?><
217 217      textarea>
218 218  </p>
219 219  <label_for="form_shield_py3">Shield_(for_Python_3)</label>
220 220  <span class="form_comment_clear">Press_"esc"_to_enable/disable_tabindent</span><br>
221 221  <textarea_id="form_shield_py3"_name="shield_py3"_rows="15"_class="sharif_input_add_text_clear"><?=$_$shield_py3_?></

```

```

221     textareax>
222     </p>
223     <p class="input_p">
224     <br>
225     <input type="submit" value="Save Changes" class="sharif_input"/>
226     </p>
227   </form>
228 <?=$this->endSection()?>

```

Kode C.15: users.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: users.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-user<?= $this->endSection()?>
8 <?= $this->section('title') ?>Users<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Users<?= $this->endSection() ?>
10 <?= $this->section('title_menu') ?>
11   <span class="title_menu_item"><a href="https://github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/users.md" target="_blank"><i class="fa_fa-question-circle_color6"></i> Help</a></span>
12   <span class="title_menu_item"><a href="?=_site_url('users/add')?>"><i class="fa_fa-plus_color11"></i> Add Users</a></span>
13   <span class="title_menu_item"><a href="?=_site_url('users/list_excel')?>"><i class="fa_fa-download_color9"></i> Excel</a></span>
14 <?= $this->endSection()?>
15 <?= $this->section('main_content') ?>
16 <!-- <php if ($deleted_user): ?>
17   <p class="shj_ok">User deleted successfully.</p>
18 <php endif ?>
19 <php if ($deleted_submissions): ?>
20   <p class="shj_ok">Submissions of selected user deleted successfully.</p>
21 <php endif ?> -->
22 <div style="height:15px"></div>
23 <table class="sharif_table">
24   <thead>
25     <tr>
26       <th>#</th>
27       <th>User ID</th>
28       <th>Username</th>
29       <th>Display Name</th>
30       <th>Email</th>
31       <th>Role</th>
32       <th>First Login</th>
33       <th>Last Login</th>
34       <th>Actions</th>
35     </tr>
36   </thead>
37   <?php foreach ($users as $index => $user): ?>
38     <tr data-id="?=_user['id']?>">
39       <td>?=> $index + 1 ?</td>
40       <td>?=> $user['id'] ?</td>
41       <td id="un"?>?=> $user['username'] ?</td>
42       <td dir="auto"?>?=> $user['display_name'] ?</td>
43       <td>?=> $user['email'] ?</td>
44       <td>?=> $user['role'] ?</td>
45       <td>?=> $user['first_login_time'] ? $user['first_login_time'] : 'Never' ?</td>
46       <td>?=> $user['last_login_time'] ? $user['last_login_time'] : 'Never' ?</td>
47       <td>
48         <a title="Edit User ?=_user['username']?" href="?=_site_url('profile/'.$user['id'])?>"><i class="fa_fa-pencil fa-lg_color9?>"</i></a>
49         <a title="Submissions User ?=_user['username']?" href="?=_site_url('submissions/all/user/'.$user['username'])?>"><i class="fa_fa-bars_fa-lg_color12?>"</i></a>
50         <button class="delete_user"><span title="Delete User ?=_user['username']?" class="pointer"><i title="Delete User" class="fa_fa-times_fa-lg_color2?>"</i></span></button>
51         <button class="delete_submissions"><span title="Delete Submissions ?=_user['username']?" class="pointer"><i class="fa_fa-times-circle_fa-lg_color1?>"</i></span></button>
52       </td>
53     </tr>
54   <?php endforeach ?>
55 </table>
56 <?= $this->endSection()?>

```

Kode C.16: edit_problem_plain.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: edit_problem_plain.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6
7 <?= $this->extend('templates/base') ?>
8 <?= $this->section('icon') ?>fa-edit<?= $this->endSection()?>
9 <?= $this->section('title') ?>Edit Problem Description (Plain HTML)<?= $this->endSection() ?>
10 <?= $this->section('head_title') ?>Edit Problem Description (Plain HTML)<?= $this->endSection() ?>
11
12
13
14 <?= $this->section('other_assets') ?>
15 <script type='text/javascript' src="?=_base_url('assets/js/tabcollapse.min.js')?>"></script>
16 <script>
17   $(document).ready(function(){

```

```

18     tabOverride.set(document.getElementById('md_editor'));
19   });
20 </script>
21 <?= $this->endSection() ?>
22
23
24 <?= $this->section('main_content') ?>
25 <p>
26   Assignment <?= $description_assignment['id'] ?> (<span dir="auto"><?= $description_assignment['name'] ?></span>)<br>
27   Problem <?= $problem['id'] ?>
28 </p>
29 <p>
30   <i class="fa fa-warning_color3"></i>
31   When you edit as html, the markdown code will be removed.
32 </p>
33 <?= form_open("problems/edit/html/".$description_assignment['id'].'.'.$problem['id']) ?>
34 <p class="input_p">
35   <textarea name="text" rows="30" cols="80" class="sharif_input" id="html_editor" aria-label="HTML_Editor"><?= $problem['
36     description'] ?></textarea>
37 </p>
38 <p class="input_p">
39   <input type="submit" value="Save" class="sharif_input"/>
40 </p>
41 </form>
42 <?= $this->endSection() ?>

```

Kode C.17: login.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: login.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->include('templates/simple_header') ?>
7
8 <?= form_open() ?>
9   <div class="box_login">
10
11     <div class="judge_logo">
12       <a href="<?= _site_url() ?>"></a>
13     </div>
14
15     <div class="login_form">
16       <div class="login1">
17         <p>
18           <label for="form_username">Username</label><br/>
19           <input id="form_username" type="text" name="username" required="required" pattern="[0-9a-zA-Z]{3,20}" title="The_
20             Username_field_must_be_between_3_and_20_characters_in_length,_and_contain_only_digits_and_lowercase_
21             letters" class="sharif_input" value="<?= _set_value('username') ?>" autofocus="autofocus"/>
22           <div class="shj_error"><?= $validationError->hasError('username') ? $validationError->getError('username') : ''
23             ?></div>
24         </p>
25         <p>
26           <label for="form_password">Password</label><br/>
27           <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The_
28             Password_field_must_be_at_least_6_characters_in_length" class="sharif_input"/>
29           <div class="shj_error"><?= $validationError->hasError('password') ? $validationError->getError('password') : ''
30             ?></div>
31           <?php if ($error): ?>
32             <div class="shj_error">Incorrect username or password.</div>
33           <?php endif ?>
34         </p>
35       <div class="login2">
36         <p style="margin:0;">
37           <?php if ($registration_enabled): ?>
38             <a href="<?= _site_url('register') ?>">Register</a> |
39           <?php endif ?>
40             <a href="<?= _site_url('login/lost') ?>">Reset Password</a>
41           <input type="submit" value="Login" id="sharif_submit"/>
42         </p>
43       </div>
44     </div>
45   </div>
46 <?= form_close() ?>
47 </body>
48 </html>

```

Kode C.18: lost.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: lost.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6
7 <?= $this->include('templates/simple_header') ?>
8
9 <?= form_open('login/lost') ?>
10  <div class="box_login">
11    <div class="judge_logo">
12      <a href="<?= _site_url() ?>"></a>

```

```

14|     </div>
15| 
16|     <div class="login_form">
17|         <div class="login1">
18|             <p>
19|                 <label for="form_email">Email</label><br/>
20|                 <input id="form_email" type="email" name="email" class="sharif_input" value="=_set_value('email')_?&gt;"/&gt;
21|                 &lt;div class="shj_error"&gt;&lt;?= isset($errors['email']) ?&gt; &lt;/div&gt;
22|             &lt;/p&gt;
23|             &lt;?php if($sent): ?&gt;
24|                 &lt;div class="shj_ok"&gt;We sent you an email containing a <a href="#"link to reset your password.</div>
25|             <?php endif ?>
26|         </div>
27|         <div class="login2">
28|             <p style="margin:0;">
29|                 <a href="=_site_url('login')_?&gt;">Login</a>
30|                 <input type="submit" value="Reset_Password" id="sharif_submit"/>
31|             </p>
32|         </div>
33|     </div>
34| 
35| </div>
36| </body>
37| </html>
38|

```

Kode C.19: register_success.php

```

1<!-- {#
2# SharIF Judge
3# file: register_success.twig
4# author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5#} -->
6<?= $this->section('title') ?>Register<?= $this->endSection() ?>
7<?= $this->include('templates/simple_header') ?>
8
9<div class="box_success">
10    <div class="judge_logo">
11        <a href="=_site_url()_?&gt;"_?>)>Login</a>
22            </p>
23        </div>
24    </div>
25</div>
26</body>
27</html>

```

Kode C.20: register.php

```

1<!-- {#
2# SharIF Judge
3# file: register.twig
4# author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5#} -->
6<?= $this->section('title') ?>Register<?= $this->endSection() ?>
7<?= $this->include('templates/simple_header') ?>
8
9<?= form_open('login/register') ?>
10<div class="box_register">
11
12    <div class="judge_logo">
13        <a href="=_site_url()_?&gt;"_?> "The Username_field_must_be_between_3_and_20_characters_in_length,_andContain_only_alpha-numeric_characters")

```

```

33         <div class="shj_error"><?= $validationError->hasError('email') ? $validationError->getError('email') : '' ?></div>
34     </p>
35     <p>
36         <label for="form_displayname">Display Name</label><br/>
37         <input id="form_displayname" type="text" name="displayname" required="required" pattern="[A-Za-z\s]+" title="The_
38             Display_Name_field_must_be_only_alphaletters" class="sharif_input" value=<?=_set_value('
39                 displayname')_?> />
40         <div class="shj_error"><?= $validationError->hasError('form_displayname') ? $validationError->getError('
41             form_displayname') : '' ?></div>
42     </p>
43     <p>
44         <label for="form_password">Password</label><br/>
45         <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The_
46             Password_field_must_be_at_least_6_characters_in_length" class="sharif_input"/>
47         <div class="shj_error"><?= $validationError->hasError('password') ? $validationError->getError('
48             password') : '' ?></div>
49     </p>
50     <p>
51         <label for="form_password_2">Password, Again</label><br/>
52         <input id="form_password_2" type="password" name="password_again" required="required" pattern=".{6,200}" title="The_
53             Password_Confirmation_field_must_be_at_least_6_characters_in_length" class="sharif_input"/>
54         <div class="shj_error"><?= $validationError->hasError('password_again') ? $validationError->getError('
55             password_again') : '' ?></div>
56     </p>
57 </div>
58 </div>
59 </div>
60 </form>
61 </body>
62 </html>
```

Kode C.21: reset_password.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: reset_password.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->include('templates/simple_header') ?>
7
8 <?= form_open('login/reset/'.$key) ?>
9 <div class="box_login">
10
11     <div class="judge_logo">
12         <a href=<?=_site_url('_?>"><img src=<?=_base_url('assets/images/banner.png')_?>"/></a>
13     </div>
14
15     <div class="login_form">
16         <div class="login1">
17             <p>
18                 <label for="form_password">New Password:</label><br/>
19                 <input id="form_password" type="password" name="password" required="required" pattern=".{6,200}" title="The_
20                     Password_field_must_be_at_least_6_characters_in_length" class="sharif_input"/>
21                 <div class="shj_error"><?= $validationError->hasError('password') ? $validationError->getError('password') : '' ?></div>
22             </p>
23
24             <p>
25                 <label for="form_password_2">New Password, Again:</label><br/>
26                 <input id="form_password_2" type="password" name="password_again" required="required" pattern=".{6,200}" title=
27                     ="The_Password_Confirmation_field_must_be_at_least_6_characters_in_length" class="sharif_input"/>
28                 <div class="shj_error"><?= $validationError->hasError('password_again') ? $validationError->getError('password
29                     ') : '' ?></div>
30             </p>
31             <?php if ($reset): ?>
32                 <div class="shj_ok">Login with your new password!</div>
33             <?php endif ?>
34         </div>
35         <div class="login2">
36             <p style="margin:0;">
37                 <a href=<?=_site_url('login')_?>">Login</a>
38                 <input type="submit" value="Set_Password" id="sharif_submit"/>
39             </p>
40         </div>
41     </div>
42 </form>
43 </body>
44 </html>
```

Kode C.22: assignments.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: assignments.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
```

```

6  <?= $this->extend('templates/base') ?>
7  <?= $this->section('icon') ?>fa-folder-open<?= $this->endSection() ?>
8  <?= $this->section('title') ?>Assignments<?= $this->endSection() ?>
9  <?= $this->section('head_title') ?>Assignments<?= $this->endSection() ?>
10 <?= $this->section('title_menu') ?>
11 <?php if ($user->level >= 2): ?>
12 <span class="title_menu_item"><a href="= _site_url('assignments/add') ?&gt;"&gt;&lt;i class="fa_fa-plus_color8"&gt;&lt;/i&gt; Add&lt;/a&gt;&lt;/span&gt;
13 &lt;?php endif ?&gt;
14 &lt;?= $this-&gt;endSection() ?&gt;
15
16 &lt;?php $this-&gt;section('main_content') ?&gt;
17 &lt;?php $msgclasses = ['success'=&gt; 'shj_g', 'notice'=&gt; 'shj_o', 'error'=&gt; 'shj_r'] ?&gt;
18 &lt;?php foreach ($messages as $message): ?&gt;
19   &lt;p class="<?= $msgclasses[$message['type']] ?&gt;"&gt;&lt;?= $message['text'] ?&gt;&lt;/p&gt;
20 &lt;?php endforeach ?&gt;
21
22 &lt;?php if (count($all_assignments) == 0): ?&gt;
23   &lt;p style="text-align:center;"&gt;Nothing to show...&lt;/p&gt;
24 &lt;?php else: ?&gt;
25 &lt;br/&gt;
26 &lt;table class="sharif_table"&gt;
27   &lt;thead&gt;
28     &lt;tr&gt;
29       &lt;th&gt;Select&lt;/th&gt;
30       &lt;th&gt;Name&lt;/th&gt;
31       &lt;th&gt;Problems&lt;/th&gt;
32       &lt;th&gt;Submissions&lt;/th&gt;
33       &lt;th&gt;Coefficient&lt;/th&gt;
34       &lt;th&gt;Start Time&lt;/th&gt;
35       &lt;th&gt;Finish Time&lt;/th&gt;
36       &lt;th&gt;Status&lt;/th&gt;
37       &lt;th&gt;PDF&lt;/th&gt;
38       &lt;th&gt;Actions&lt;/th&gt;
39     &lt;?php if ($user-&gt;level &gt; 0): ?&gt;
40       &lt;th&gt;&lt;/th&gt;
41     &lt;?php endif ?&gt;
42   &lt;/tr&gt;
43 &lt;/thead&gt;
44 &lt;?php foreach (array_reverse($all_assignments) as $item): ?&gt;
45   &lt;tr&gt;
46     &lt;td&gt;&lt;i tabindex="0" class="pointer_select_assignment_fa_&lt;?= $item['id'] ?&gt;_fa-selected_assignment['id']_fa-check-square_color6_fa-square-o_fa-2x" data-id="<?= $item['id'] ?&gt;"&gt;&lt;/i&gt;&lt;/td&gt;
47     &lt;td dir="auto"&gt;<?= $item['name'] ?&gt;&lt;/td&gt;
48     &lt;td&gt;&lt;a href="<?= _site_url('problems/' . $item['id']) ?&gt;"&gt;&lt;?= $item['problems'] ?&gt; problem&lt;?= $item['problems'] != 1 ? 's' : ''&gt;&lt;/a&gt;&lt;/td&gt;
49     &lt;td&gt;<?= $item['total_submits'] ?&gt; submission&lt;?= $item['total_submits'] != 1 ? 's' : ''&gt;&lt;/td&gt;
50     &lt;td&gt;
51       &lt;?php if ($item['finished']): ?&gt;
52         &lt;span style="color: red;"&gt;Finished&lt;/span&gt;
53       &lt;?php else: ?&gt;
54         &lt;?= $item['coefficient'] ?&gt; %
55       &lt;?php endif ?&gt;
56     &lt;/td&gt;
57     &lt;td&gt;<?= $item['start_time'] ?&gt;&lt;/td&gt;
58     &lt;td&gt;<?= $item['finish_time'] ?&gt;&lt;/td&gt;
59     &lt;td&gt;
60       &lt;?php if ($item['open']): ?&gt;
61         &lt;span style="color: green;"&gt;Open&lt;/span&gt;
62       &lt;?php else: ?&gt;
63         &lt;span style="color: red;"&gt;Close&lt;/span&gt;
64       &lt;?php endif ?&gt;
65     &lt;/td&gt;
66     &lt;td&gt;
67       &lt;a href="<?= _site_url('assignments/pdf/' . $item['id']) ?&gt;"&gt;&lt;img src="<?= _base_url('assets/images/pdf.svg') ?&gt;" aria-label="Download_PDF_For_Assignment_&lt;?= $item['name'] ?&gt;"&gt;&lt;/a&gt;
68     &lt;/td&gt;
69     &lt;?php if ($user-&gt;level &gt; 0): ?&gt;
70     &lt;td&gt;
71       &lt;?php if ($user-&gt;level &gt;= 2): ?&gt;
72         &lt;a href="<?= _site_url('assignments/downloadtestsdesc/' . $item['id']) ?&gt;"&gt;&lt;i title="Download_Tests_and_Descriptions" class="fa_fa-cloud-download_fa-lg_color11"&gt;&lt;/i&gt;&lt;/a&gt;
73       &lt;?php endif ?&gt;
74       &lt;?php if ($user-&gt;level &gt;= 1): ?&gt;
75         &lt;a href="<?= _site_url('assignments/download_submissions/by_user/' . $item['id']) ?&gt;"&gt;&lt;i title="Download_Final_Submissions_(by_user)" class="fa_fa-download_fa-lg_color12"&gt;&lt;/i&gt;&lt;/a&gt;
76         &lt;a href="<?= _site_url('assignments/download_submissions/by_problem/' . $item['id']) ?&gt;"&gt;&lt;i title="Download_Final_Submissions_(by_problem)" class="fa_fa-download_fa-lg_color2"&gt;&lt;/i&gt;&lt;/a&gt;
77       &lt;?php endif ?&gt;
78       &lt;?php if ($user-&gt;level &gt;= 2): ?&gt;
79         &lt;a href="<?= _site_url('moss/' . $item['id']) ?&gt;"&gt;&lt;i title="Detect_Similar_Codes" class="fa_fa-shield_fa-lg_color7"&gt;&lt;/i&gt;&lt;/a&gt;
80       &lt;?php endif ?&gt;
81       &lt;?php if ($user-&gt;level &gt;= 2): ?&gt;
82         &lt;a href="<?= _site_url('assignments/edit/' . $item['id']) ?&gt;"&gt;&lt;i title="Edit" class="fa_fa-pencil_fa-lg_color3"&gt;&lt;/i&gt;&lt;/a&gt;
83       &lt;?php endif ?&gt;
84       &lt;?php if ($user-&gt;level &gt;= 2): ?&gt;
85         &lt;a href="<?= _site_url('assignments/delete/' . $item['id']) ?&gt;"&gt;&lt;i title="Delete" class="fa_fa-times_fa-lg_color1"&gt;&lt;/i&gt;&lt;/a&gt;
86       &lt;?php endif ?&gt;
87     &lt;/td&gt;
88     &lt;?php endif ?&gt;
89   &lt;/tr&gt;
90 &lt;?php endforeach ?&gt;
91 &lt;/table&gt;
92 &lt;?php endif ?&gt;
93 &lt;?php $this-&gt;endSection() ?&gt;
</pre

```

Kode C.23: dashboard.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: dashboard.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-dashboard<?= $this->endSection()?>
8 <?= $this->section('title') ?>Dashboard<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Dashboard<?= $this->endSection() ?>
10
11 <?= $this->section('other_assets')?>
12 <link rel='stylesheet' type='text/css' href='<?=base_url('assets/fullcalendar/fullcalendar.css')?>' />
13 <script type='text/javascript' src='<?=base_url('assets/fullcalendar/fullcalendar.min.js')?>'></script>
14 <link rel='stylesheet' type='text/css' href='<?=base_url('assets/gridster/jquery.gridster.css')?>' />
15 <script type='text/javascript' src='<?=base_url('assets/gridster/jquery.gridster.min.js')?>'></script>
16 <script type='text/javascript' src='<?=base_url('assets/js/jquery.autoellipsis-1.0.10.min.js')?>'></script>
17 <script>
18 $(document).ready(function () {
19     $('#calendar').fullCalendar({
20         timeFormat: 'HH:mm_{_}-HH:mm',
21         editable: false,
22         height: 280,
23         firstDay: <?= $week_start ?>,
24         events: [
25             <?php $colors = ['#812C8C', '#FF750D', '#2C578C', '#013440', '#A6222C', '#42758C', '#02A300', '#BA6900']; ?>
26             <?php foreach ($all_assignments as $idx => $assignment): ?>
27                 <?php if ($assignment['archived_assignment'] == '0'): ?>
28                     {id: <?= $assignment['id'] ?>, title: '<?=$assignment['name']?>', start: '<?=$assignment['start_time']?>',
29                      end: '<?=$assignment['finish_time']?>', allDay: false, color: '<?=$colors[($idx) % count($colors)]?>'},
30                 <?php endif; ?>
31             <?php endforeach; ?>
32         ],
33     });
34     var gridster = $(".gridster_ul").gridster({
35         widget_margins: [10, 10],
36         widget_base_dimensions: [350, 350],
37         serialize_params: function ($w, wgd) {
38             return {
39                 r: wgd.row,
40                 c: wgd.col,
41                 x: wgd.size_x,
42                 y: wgd.size_y
43             }
44         },
45         draggable: {
46             handle: '.widget_title',
47             stop: function (event, ui) { // send widget positions to server for saving in database
48                 var positions = JSON.stringify(gridster.serialize());
49                 $.post(
50                     "<?=_site_url('/dashboard/widget_positions')?>",
51                     {
52                         positions: positions,
53                         shj_csrf_token: shj.csrf_token
54                     }
55                 );
56             }
57         }.data('gridster');
58     $('.notif_text').ellipsis();
59 });
60 </script>
61 <?= $this->endSection()?>
62
63 <?= $this->section('main_content')?>
64 <?php foreach ($errors as $error){ ?>
65     <p class="shj_error"><?= $error ?></p>
66 <?php } ?>
67 <div style="height:_15px;"></div>
68 <div class="gridster">
69     <?php $i = 0; ?>
70     <ul>
71         <li data-row="<?=_isset($wp[0])?_&$wp[0]['r']:'_1'?>" data-col="<?=_isset($wp[0])?_&$wp[0]['c']:'_1'?>" data-sizex="<?=_isset($wp[0])?_&$wp[0]['x']:'_1'?>" data-sizey="<?=_isset($wp[0])?_&$wp[0]['y']:'_1'?>">
72             <div class="shj_widget">
73                 <div class="widget_title"><i class="fa_fa-calendar-o_fa-lg_color10"></i> Calendar</div>
74                 <div class="widget_scrollable_scroll-wrapper">
75                     <div class="scroll-content">
76                         <div class="widget_contents_container" id='calendar'></div>
77                     </div>
78                 </div>
79             </div>
80         </li>
81         <li data-row="<?=_isset($wp[1])?_&$wp[1]['r']:'_1'?>" data-col="<?=_isset($wp[1])?_&$wp[1]['c']:'_2'?>" data-sizex="<?=_isset($wp[1])?_&$wp[1]['x']:'_1'?>" data-sizey="<?=_isset($wp[1])?_&$wp[1]['y']:'_1'?>">
82             <div class="shj_widget">
83                 <div class="widget_title"><i class="fa_fa-bell-o_fa-lg_color2"></i> Latest Notifications
84                 <?php if ($user->level >= 2): ?>
85                     <a title="New Notification" href="<?=_site_url('notifications/add')?>" class="pull-right">
86                         <i class="fa_fa-plus_color11"></i>
87                     </a>
88                 <?php endif ?>
89             </div>
90             <div class="widget_scrollable_scroll-wrapper">
91                 <div class="scroll-content">
92                     <div class="widget_contents_container">

```

```

96|         <?php if (count($notifications) == 0): ?>
97|             <p style="text-align:center;">Nothing yet...</p>
98|         </php endif ?>
99|         <?php foreach ($notifications as $notification): ?>
100|             <div class="notif" id="number<=?$_notification['id']_?>" data-id="<=?$_notification['id']_?>">
101|                 <div class="notif_title" dir="auto">
102|                     <span class="anchor_ttl_n"><=? $notification['title'] ?></span>
103|                     <span class="notif_meta" dir="ltr">
104|                         <?= $notification['time'] ?>
105|                             <?php if ($user->level >= 2): ?>
106|                                 <span class="anchor_edt_n">Edit</span>
107|                                 <span class="anchor_del_n">Delete</span>
108|                             <?php endif ?>
109|                         </span>
110|                     </div>
111|                     <div class="notif_text_latest" dir="auto"><=? $notification['text'] ?></div>
112|                 </div>
113|             <?php endforeach ?>
114|         </div>
115|     </div>
116|     </div>
117|     </div>
118|     </li>
119|     <li data-row="<?=_isset($wp[2])_?_&$wp[2]['r']_?_2_?>" data-col="<?=_isset($wp[2])_?_&$wp[2]['c']_?_1_?>" data-sizey="<?=_isset($wp[2])_?_&$wp[2]['y']_?_1_?>">
120|         <?=_isset($wp[2])_?_&$wp[2]['x']_?_1_?>
121|     </li>
122| </ul>
123| </div>
124| <?= $this->endSection() ?>

```

Kode C.24: halloffame.php

```

1<!-- {#
2# SharIF Judge
3# file: halloffame.twig
4# author: Stillmen Vallian <stillmen.v@gmail.com>
5#} -->
6<?= $this->extend('templates/base') ?>
7<?= $this->section('icon') ?>fa fa-list-alt fa-lg<?= $this->endSection() ?>
8<?= $this->section('title') ?>Hall of Fame<?= $this->endSection() ?>
9<?= $this->section('head_title') ?>Hall of Fame<?= $this->endSection() ?>
10<?= $this->section('other_assets')?>
11<link rel='stylesheet' type='text/css' href='<?=_base_url("assets/snippet/jquery.snippet.css")_?>' />
12<link rel='stylesheet' type='text/css' href='<?=_base_url("assets/snippet/themes/github.css")_?>' />
13<script type='text/javascript' src="<?=_base_url("assets/snippet/jquery.snippet.js")_?>"></script>
14<link rel='stylesheet' type='text/css' href='<?=_base_url("assets/reveal/reveal.css")_?>' />
15<script type='text/javascript' src="<?=_base_url("assets/reveal/jquery.reveal.js")_?>"></script>
16<?= $this->endSection() ?>
17
18
19<?= $this->section('main_content') ?>
20<div style="height:15px"></div>
21<table class="sharif_table">
22    <thead>
23        <tr>
24            <th>Rank</th>
25            <th>Username</th>
26            <th>Display Name</th>
27            <th>Total Score</th>
28        </tr>
29    </thead>
30    <?php $tempTotalScore = 0 ?>
31    <?php $tempLoop = 0 ?>
32    <?php $count = 0 ?>
33    <?php foreach($hofs as $index => $hof): ?>
34    <?php $index += 1 ?>
35        <tr class="hof_details" style="cursor:pointer;">
36            <?php if ($index == 1): ?>
37                <td>1</td>
38                <?php $tempTotalScore = $hof['totalscore'] ?>
39                <?php $tempLoop = $tempLoop+1 ?>
40            <?php elseif ($tempTotalScore == $hof['totalscore']): ?>
41                <td><?= $tempLoop ?></td>
42                <?php $count = $count+1 ?>
43            <?php else: ?>
44                <?php $tempTotalScore = $hof['totalscore'] ?>
45                <?php $tempLoop = $count+$tempLoop+1 ?>
46                <td><?= $tempLoop ?></td>
47                <?php $count = 0 ?>
48            <?php endif ?>
49                <td class="username"><?= $hof['username'] ?></td>
50                <td class="display_name"><?= $hof['display_name'] ?></td>
51                <td><?= $hof['totalscore'] ?></td>
52            </tr>
53        <?php endforeach ?>
54    </table>
55    <?= $this->endSection() ?>
56
57    <?= $this->section('body_end') ?>
58    <div id="shj_modal" class="reveal-modal_xlarge">
59        <div class="modal_inside">
60            <div style="text-align:center;">Loading<br></div>
61        </div>
62        <a class="close-reveal-modal">#215;</a>
63    </div>
64    <?= $this->endSection() ?>

```

Kode C.25: notifications.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: notifications.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-bell<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Notifications<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Notifications<?= $this->endSection() ?>
10 <?= $this->section('title_menu') ?>
11   <?php if ($user->level >= 2): ?>
12     <span class="title_menu_item"><a href="=site_url('notifications/add')?&gt;"&gt;&lt;i class="fa_fa-plus_color10"&gt;&lt;/i&gt; New&lt;/a&gt;&lt;/span&gt;
13   &lt;/?php endif ?&gt;
14 &lt;?= $this-&gt;endSection() ?&gt;
15
16
17
18
19 &lt;?= $this-&gt;section('main_content') ?&gt;
20 &lt;?php if (count($notifications) == 0): ?&gt;
21   &lt;p style="text-align:center;"&gt;Nothing yet...&lt;/p&gt;
22 &lt;?php endif ?&gt;
23 &lt;?php foreach ($notifications as $notification): ?&gt;
24   &lt;div class="notif" id="number&lt;?=$notification['id']?&gt;" data-id="<?=$notification['id']?&gt;"
25     <div class="notif_title" dir="auto">
26       <span class="anchor_ttl_n"><?= $notification['title'] ?></span>
27       <div class="notif_meta">
28         <?= $notification['time'] ?>
29         <?php if ($user->level >= 2): ?>
30           <span class="anchor_edit_n">Edit</span>
31           <span class="anchor_del_n">Delete</span>
32         <?php endif ?>
33       </div>
34     </div>
35     <div class="notif_text" dir="auto">
36       <?= $notification['text'] ?>
37     </div>
38   </div>
39 <?php endforeach ?>
40 <?= $this->endSection() ?>

```

Kode C.26: problems.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: problems.twig
4 # author: Kelvin Ng <kelvin9302104@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-puzzle-piece<?= $this->endSection() ?>
8 <?= $this->section('title') ?><?= $all_problems[$problem['id']]['name'] ?><?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Problem <?= $problem['id'] ?><?= $this->endSection() ?>
10
11 <?= $this->section('other_assets') ?>
12 <link rel='stylesheet' type='text/css' href='<?=base_url("assets/snippet/jquery.snippet.css")?>' />
13 <link rel='stylesheet' type='text/css' href='<?=base_url("assets/snippet/themes/github.css")?>' />
14 <script type='text/javascript' src="<?=base_url("assets/snippet/jquery.snippet.js")?>"></script>
15 <script>
16 $(document).ready(function(){
17   // Syntax highlighting increases the page's height, and we need to update the scroll-bar
18   $('.problem_description').resize(function(){
19     $('.scroll-wrapper').nanoScroller();
20   });
21   // Fix text directions for rtl text
22   $.each($('.problem_description[dir="auto"]'), function(i, element){
23     if (getComputedStyle(element).direction == 'rtl')
24     {
25       $(element).css('direction', 'rtl');
26       $(element).parent('ul,ol').css('direction', 'rtl');
27     }
28   });
29   // Syntax highlighting
30   $('pre_code.language-c').parent().snippet('c', {style: shj.color_scheme});
31   $('pre_code.language-cpp').parent().snippet('cpp', {style: shj.color_scheme});
32   $('pre_code.language-python').parent().snippet('python', {style: shj.color_scheme});
33   $('pre_code.language-java').parent().snippet('java', {style: shj.color_scheme});
34 });
35 </script>
36 <?= $this->endSection() ?>
37
38
39
40 <?= $this->section('title_menu') ?>
41 <?php if ($user->level >= 2): ?>
42   <span class="title_menu_item"><a href="=site_url("problems/edit/md/".$description_assignment['id'].'/.'.$problem['id'])?&gt;"&gt;&lt;i class="fa_fa-pencil_color2"&gt;&lt;/i&gt; Edit Markdown&lt;/a&gt;&lt;/span&gt;
43   &lt;span class="title_menu_item"&gt;&lt;a href="<?=site_url("problems/edit/html/".$description_assignment['id'].'/.'.$problem['id'])?&gt;"&gt;&lt;i class="fa_fa-pencil_color10"&gt;&lt;/i&gt; Edit HTML&lt;/a&gt;&lt;/span&gt;
44   &lt;span class="title_menu_item"&gt;&lt;a href="<?=site_url("problems/edit/plain/".$description_assignment['id'].'/.'.$problem['id'])?&gt;"&gt;&lt;i class="fa_fa-pencil_color8"&gt;&lt;/i&gt; Edit Plain HTML&lt;/a&gt;&lt;/span&gt;
45 &lt;?php if ($problem['has_pdf']): ?&gt;
46   &lt;span class="title_menu_item"&gt;&lt;a href="<?=site_url("assignments/pdf/".$description_assignment['id'].'/.'.$problem['id'])?&gt;"&gt;&lt;i class="fa_fa-download_color1"&gt;&lt;/i&gt; PDF&lt;/a&gt;&lt;/span&gt;
47 &lt;?php endif ?&gt;
48 &lt;?php endif ?&gt;
</pre

```

```

49 <?= $this->endSection() ?>
50
51
52
53 <?= $this->section('main_content') ?>
54 <div class="problem_description">
55   <?= $problem['description'] ?>
56 </div>
57
58 <div id="right_sidebar">
59
60   <div class="problems_widget">
61     <p dir="auto"><i class="fa fa-file-text_fa-lg_color9"></i> <?= $description_assignment['name'] ?></p>
62
63     <?php if (count($all_problems) == 0): ?>
64       <p style="text-align:center;">Nothing to show...</p>
65     <?php endif ?>
66
67     <table class="sharif_table">
68       <thead>
69         <tr>
70           <th rowspan="2">#</th>
71           <th rowspan="2">Problem</th>
72           <th rowspan="2">Score</th>
73           <th rowspan="2">Upload<br>Only</th>
74         </tr>
75       </thead>
76       <?php foreach($all_problems as $one_problem): ?>
77         <tr <?= $problem['id'] == $one_problem['id'] ? ' class="hl"' : '' ?>
78           <td <?= $one_problem['id'] ?></td>
79           <td>
80             <a dir="auto" href="<?= _site_url("problems/".$description_assignment['id']).'/'.$one_problem['id']."'><?= $one_problem['name'] ?></a>
81           </td>
82           <td><?= $one_problem['score'] ?></td>
83           <td><?= $one_problem['is_upload_only'] ? 'Yes' : 'No' ?></td>
84         </tr>
85       </?php endforeach ?>
86     </table>
87   </div>
88
89   <?php if ($can_submit): ?>
90   <div class="problems_widget">
91     <p><i class="fa fa-upload_fa-lg_color11"></i> Submit</p>
92     <?= form_open_multipart('submit') ?>
93     <input type="hidden" name="problem" value="<?= $problem['id'] ?>"/>
94
95     <p class="input_p">
96       <select id="languages" name="language" class="sharif_input_full-width" aria-label="Select_Language">
97         <option value="0" selected="selected">-- Select Language --</option>
98         <?php foreach ($problem['allowed_languages'] as $l): ?>
99           <option value="<?= $l ?>"><?= $l ?></option>
100        <?php endforeach ?>
101      </select>
102    </p>
103    <p class="input_p">
104      <input type="file" id="file" class="sharif_input_full-width" name="userfile" aria-label="Upload_File"/>
105    </p>
106    <p class="input_p">
107      <input type="submit" value="Submit" class="sharif_input"/>
108    </p>
109  </form>
110 </div>
111 <?php endif ?>
112
113 </div>
114 <?= $this->endSection() ?> {# main_content #}

```

Kode C.27: profile.php

```

1 <!!--- {#
2 # SharIF Judge
3 # file: profile.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-user<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Profile<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Profile<?= $this->endSection() ?>
10
11 <?= $this->section('main_content')?>
12 <p class="input_p">
13 <?php if ($form_status == 'ok'): ?>
14 <div class="shj_ok">Profile updated successfully.</div>
15 <?php elseif ($form_status == 'error'): ?>
16   <div class="shj_error">Error updating profile.</div>
17 <?php endif ?>
18 </p>
19 <?= form_open('profile/'. $id) ?>
20 <p class="input_p_clear">
21   <label for="form_username" class="short2">Username:<br>
22     <span class="form_comment">You cannot change username.</span>
23   </label>
24   <input id="form_username" type="text" name="username" class="sharif_input_medium" value="<?= $edit_username ?>" disabled/>
25 </p>
26 <p class="input_p_clear">
27   <label for="form_name" class="short2">Name:</label>
28   <?php if ($lock_student_display_name == 1): ?>

```

```

29      <input id="form_name" type="text" name="display_name" class="sharif_input_medium" value="<?=$_display_name_?>" disabled/>
30      <?php else: ?>
31          <input id="form_name" type="text" name="display_name" class="sharif_input_medium" value="<?=$_display_name_?>"/>
32      <?php endif ?>
33      <div class="shj_error"><?= $validationError->hasError('display_name') ? $validationError->getError('display_name') : '' ?></div>
34  </p>
35  <p class="input_p_clear">
36      <label for="form_email" class="short2">Email:</label>
37      <input id="form_email" type="text" name="email" class="sharif_input_medium" value="<?=$_email_?>"/>
38      <div class="shj_error"><?= $validationError->hasError('email') ? $validationError->getError('email') : '' ?></div>
39  </p>
40  <p class="input_p_clear">
41      <label for="form_password" class="short2">Password:<br>
42          <span class="form_comment">If you don't want to change password, leave this blank.</span>
43      </label>
44      <input id="form_password" type="password" name="password" class="sharif_input_medium"/>
45      <div class="shj_error"><?= $validationError->hasError('password') ? $validationError->getError('password') : '' ?></div>
46  </p>
47  <p class="input_p_clear">
48      <label for="form_password_2" class="short2">Password Again:</label>
49      <input id="form_password_2" type="password" name="password_again" class="sharif_input_medium"/>
50      <div class="shj_error"><?= $validationError->hasError('password_again') ? $validationError->getError('password_again') : '' ?></div>
51  </p>
52  <?php if($_level == 3): ?>
53  <p class="input_p_clear">
54      <label for="form_role" class="short2">User Role:</label>
55      <select id="form_role" name="role" class="sharif_input">
56          <option value="admin" <?=$_role == 'admin' ?> selected="selected":><?> admin</option>
57          <option value="head_instructor" <?=$_role == 'head_instructor' ?> selected="selected":><?> head_instructor</option>
58          <option value="instructor" <?=$_role == 'instructor' ?> selected="selected":><?> instructor</option>
59          <option value="student" <?=$_role == 'student' ?> selected="selected":><?> student</option>
60      </select>
61      <div class="shj_error"><?= $validationError->hasError('role') ? $validationError->getError('role') : '' ?></div>
62  </p>
63  <?php endif ?>
64  <p class="input_p">
65      <input type="submit" value="Save" class="sharif_input"/>
66  </p>
67  </form>
68  <?= $this->endSection() ?>

```

Kode C.28: scoreboard_table.php

```

1  <!-- {#
2  # ShariF Judge
3  # file: scoreboard_table.twig
4  # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5  #} -->
6  <table class="sharif_table">
7
8  <thead>
9      <tr>
10         <th>#</th>
11         <th>Username</th>
12         <th>Name</th>
13         <?php foreach($problems as $problem): ?>
14             <th>
15                 <a dir="auto" href="<?= _site_url("problems/$assignment_id/") . $problem['id'] ?>"><?= $problem['name'] ?></a><br>
16                 <span class="tiny_text_b"><?= $problem['score'] ?></span>
17             </th>
18         <?php endforeach ?>
19         <th>
20             Total<br>
21             <span class="tiny_text_b"><?= $total_score ?></span>
22         </th>
23     </tr>
24 </thead>
25
26 <?php foreach($scoreboard['username'] as $index => $sc_username): ?>
27     <tr>
28         <td><?= $index + 1 ?></td>
29         <td><?= $sc_username ?></td>
30         <td dir="auto"><?= $names[$sc_username] ?></td>
31         <?php foreach($problems as $problem): ?>
32             <td>
33                 <?php if (isset($scores[$sc_username][$problem['id']]['score'])): ?>
34                     <?= $scores[$sc_username][$problem['id']]['score'] ?><br>
35                     <span class="tiny_text" title="Time"><?= time_hhmm($scores[$sc_username][$problem['id']]['time']) ?></span>
36                 <?php else: ?>
37                     -
38                 <?php endif ?>
39             </td>
40         <?php endforeach ?>
41         <td>
42             <span style="font-weight:bold;"><?= $scoreboard['score'][$index] ?></span>
43             <br>
44             <span class="tiny_text" title="Total_Time+_Submit_Penalty"><?= time_hhmm($scoreboard['submit_penalty'][$index]) ?></span>
45         </td>
46     </tr>
47 <?php endforeach ?>
48 </table>

```

Kode C.29: scoreboard.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: scoreboard.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-star<?= $this->endSection()?>
8 <?= $this->section('title') ?>Scoreboard<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Scoreboard<?= $this->endSection() ?>
10 <?= $this->section('main_content') ?>
11   <?php if ($user->selected_assignment['id'] == 0): ?>
12   <p>No assignment is selected.</p>
13   <?php elseif (!$user->selected_assignment['scoreboard']): ?>
14   <p>Scoreboard is disabled.</p>
15   <?php else: ?>
16     <p>Scoreboard of <span dir="auto"><?= $user->selected_assignment['name'] ?></span></p>
17     <?= $scoreboard ?>
18   <?php endif ?>
19 <?= $this->endSection() ?>

```

Kode C.30: submissions.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: submissions.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?><?= $view == 'all' ?>'fa-bars' : 'fa-map-marker' ?><?= $this->endSection() ?>
8 <?= $this->section('title') ?><?= ucfirst($view) ?> Submissions<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?><?= ucfirst($view) ?> Submissions<?= $this->endSection() ?>
10
11
12 <?= $this->section('other_assets') ?>
13   <link rel='stylesheet' type='text/css' href='<?= _base_url("assets/snippet/jquery.snippet.css") ?>'>
14   <link rel='stylesheet' type='text/css' href='<?= _base_url("assets/snippet/themes/github.css") ?>'>
15   <script type='text/javascript' src='<?= _base_url("assets/snippet/jquery.snippet.js") ?>'></script>
16
17   <link rel='stylesheet' type='text/css' href='<?= _base_url("assets/reveal/reveal.css") ?>'>
18   <script type='text/javascript' src='<?= _base_url("assets/reveal/jquery.reveal.js") ?>'></script>
19   <script type='text/javascript' src='<?= _base_url("assets/js/shj_submissions.js") ?>'></script>
20 <?= $this->endSection() ?>
21
22
23
24 <?= $this->section('title_menu') ?>
25   <span class="title_menu_item">
26     <a href='<?= $excel_link ?>'><i class="fa_fa-download_color2" style="color: #007bff;"></i> Excel</a>
27   </span>
28   <?php if ($filter_user): ?>
29   <span class="title_menu_item">
30     <a href='<?= site_url('submissions/' . $view . ($filter_problem ? '/problem/' . $filter_problem : '')) ?>'>
31       <i class="fa_fa-filter_color1" style="color: #007bff;"></i> Remove Username Filter</a>
32   </span>
33   <?php endif ?>
34   <?php if ($filter_problem): ?>
35   <span class="title_menu_item">
36     <a href='<?= site_url('submissions/' . $view . ($filter_user ? '/user/' . $filter_user : '')) ?>'>
37       <i class="fa_fa-filter_color4" style="color: #007bff;"></i> Remove Problem Filter</a>
38   </span>
39   <?php endif ?>
40 <?= $this->endSection() ?>
41
42
43
44
45 <?= $this->section('main_content') ?>
46
47 <p><?= ucfirst($view) ?> Submissions of <span dir="auto"><?= $user->selected_assignment['name'] ?></span></p>
48 <?php if ($view == 'all'): ?>
49   <p><i class="fa_fa-warning_color3" style="color: #dc3545;"></i> You cannot change your final submissions after assignment finishes.</p>
50 <?php endif ?>
51 <table class="sharif_table">
52   <thead>
53     <tr>
54       <?php if ($view == 'all'): ?>
55         <th width="1%" rowspan="2" style="text-align: center;">Final</th>
56       <?php endif ?>
57       <?php if ($user->level > 0): ?>
58         <th width="3%" rowspan="2" style="text-align: center;">ID</th>
59       <?php if ($view == 'all'): ?>
60         <th width="6%" rowspan="2" style="text-align: center;">Username</th>
61       <?php else: ?>
62         <th width="2%" rowspan="2" style="text-align: center;">#</th>
63       <th width="3%" rowspan="2" style="text-align: center;">Name</th>
64       <?php endif ?>
65       <th width="14%" rowspan="2" style="text-align: center;">Problem</th>
66       <th width="14%" rowspan="2" style="text-align: center;">Submit Time</th>
67       <th width="4%" rowspan="2" style="text-align: center;">Score</th>
68       <th width="1%" rowspan="2" style="text-align: center;">Language</th>
69       <th width="6%" rowspan="2" style="text-align: center;">Status</th>
70       <th width="6%" rowspan="2" style="text-align: center;">Code</th>
71       <th width="6%" rowspan="2" style="text-align: center;">Log</th>
72       <?php if ($user->level >= 2): ?>
73         <th width="1%" rowspan="2" style="text-align: center;">Actions</th>
74       <?php endif ?>
75     </tr>

```

```

76        <?php endif ?>
77    </tr>
78    <tr>
79        <th width="5%" class="score">Score</th>
80        <th width="5%" class="score">Delay<br>%</th>
81        <th width="5%" class="score">Final Score</th>
82    </tr>
83    <?php else: ?>
84        <th width="10%" rowspan="2">Problem</th>
85        <th width="30%" rowspan="2">Submit Time</th>
86        <th width="7%" colspan="3">Score</th>
87        <th width="1%" rowspan="2">Language</th>
88        <th width="30%" rowspan="2">Status</th>
89        <th width="15%" rowspan="2">Code</th>
90    </tr>
91    <tr>
92        <th width="7%" class="score">Score</th>
93        <th width="7%" class="score">Delay<br>%</th>
94        <th width="7%" class="score">Final Score</th>
95    </tr>
96    <?php endif ?>
97 </thead>
98 <?php $i=0 ?>
99 <?php $j=0 ?>
100 <?php $un=' '?>
101 <?php foreach ($submissions as $submission): ?>
102     <?php $i = $i+1 ?>
103     <?php if ($submission['username'] != $un )?:?>
104         <?php $j = $j+1 ?>
105     <?php endif ?>
106     <?php $un = $submission['username'] ?>
107     <tr data-u=<?=$submission['username']_?> data-a=<?=$submission['assignment']_?> data-p=<?=$submission['problem']_?>
108         <?php data-s='<?=$submission['submit_id']_?>' <?php if ($view == 'final' && ($j % 2 == 0)):?:?>class="hl"<?php endif ?>>>
109     <?php if ($view == 'all'): ?>
110         <td>
111             <i tabindex="0" aria-label="Set_Final" role="checkbox" aria-checked=<?=$submission['is_final']_?_>'true'_?_>'false
112                 '_?> class="pointer_set_final_fa_<?=$submission['is_final']_?_>fa-check-circle-o_color11'_?_>fa-circle-o_<?=$fa-2x'_?>/</i>
113         </td>
114     <?php endif ?>
115     <?php if ($user->level > 0): ?>
116         <td>
117             <?php if ($view == 'all'): ?>
118                 <td><?= ($page_number-1)*$per_page+$i ?></td>
119                 <td><?= $submission['submit_id'] ?></td>
120             <?php endif ?>
121             <td><a href=<?=_site_url('submissions/'.$view.'/'.$user.'/'.$submission['username']).($filter_problem?'/problem/'.$filter_problem:_'?')_?>>?>= $submission['username'] ?></a></td>
122             <td dir="auto"><?=$submission['name'] ?></td>
123         <?php endif ?>
124         <td><a href=<?=_site_url('submissions/'.$view.'($filter_user?/'.$user.'/'.$filter_user:_'?').'/problem/'.$submission['problem'])_?>" title=<?=$problems[$submission['problem']][name]_?>><?=$submission['problem'] ?></a></td>
125         <td><?=$submission['time'] ?></td>
126         <td><?=$submission['pre_score'] ?></td>
127         <td>
128             <span class="tiny_text" <?= $submission['delay'] > 0 ? 'style="color:red;"' : '' ?>
129             <?php if ($submission['delay'] <= 0): ?>
130                 No Delay
131             <?php else: ?>
132                 <span title="HH:MM"><?= time_hhmm($submission['delay']) ?></span>
133             <?php endif ?>
134             </span><br>
135             <?= $submission['coefficient'] ?>%
136         </td>
137         <td style="font-weight:bold;"><?=$submission['final_score'] ?> </td>
138         <td><?=$submission['language'] ?></td>
139         <td class="status">
140             <?php if ($submission['status'] == 'Uploaded'): ?>
141                 Uploaded
142             <?php else: ?>
143                 <?php if (strtolower($submission['status']) == 'pending'): ?>
144                     <?php $submission_class = 'btn' ?>
145                 <?php elseif (strtolower($submission['status']) == 'score'): ?>
146                     <?php $submission_class = ($submission['fullmark'] ? 'btn_shj-green' : 'btn_shj-red') ?>
147                 <?php else: ?>
148                     <?php $submission_class = 'btn_shj-blue' ?>
149                 <?php endif ?>
150                 <div tabindex="0" class=<?=$submission_class_?> data-type="result" >
151                     <?php if ($submission['status'] == 'SCORE'): ?>
152                         <?= $submission['final_score'] ?>
153                     <?php else: ?>
154                         <?= $submission['status'] ?>
155                     <?php endif ?>
156                 </div>
157             <?php endif ?>
158         </td>
159         <td>
160             <?php if ($submission['file_type'] == 'zip' || $submission['file_type'] == 'pdf' || $submission['file_type'] == 'txt'): ?>
161                 <div tabindex="0" class="btn_shj-orange" data-type="download">Download</div>
162             <?php else: ?>
163                 <div tabindex="0" class="btn_shj-orange" data-type="code" >Code</div>
164             <?php endif ?>
165         </td>
166         <?php if ($user->level > 0): ?>
167             <td>
```

```

168         <?php if ($submission['status'] == 'Uploaded'): ?>
169             ...
170             <?php else: ?>
171                 <div tabindex="0" class="btn" data-type="log" >Log</div>
172             <?php endif ?>
173         </td>
174         <?php endif ?>
175         <?php if ($user->level >= 2): ?>
176             <td>
177                 <div tabindex="0" class="shj_rejudge_pointer"><i class="fa fa-refresh fa-lg color10"></i></div>
178             </td>
179         <?php endif ?>
180     </tr>
181     <?php endforeach ?>
182 </table>
183
184 <p>
185 <?= $pagination ?>
186 </p>
187
188 <?= $this->endSection() ?>
189
190
191 <?= $this->section('body_end') ?>
192 <div id="shj_modal" class="reveal-modal_xlarge">
193     <div class="modal_inside">
194         <div style="text-align:center;">Loading<br>
198 <?= $this->endSection() ?>

```

Kode C.31: submit.php

```

1 <!!-- {#
2 # SharIF Judge
3 # file: submit.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->extend('templates/base') ?>
7 <?= $this->section('icon') ?>fa-location-arrow<?= $this->endSection() ?>
8 <?= $this->section('title') ?>Submit<?= $this->endSection() ?>
9 <?= $this->section('head_title') ?>Submit<?= $this->endSection() ?>
10
11
12
13 <?= $this->section('other_assets') ?>
14 <link rel="stylesheet" type="text/css" href="=base_url('../assets/styles/submit.css')?"/&gt;
15     &lt;script&gt;
16         shj.p={};
17         &lt;?= $problems_js ?&gt;;
18         $ajax({
19             url: "<?=site_url('assignments/pdfCheck/'.$user-&gt;selected_assignment['id'])?&gt;",
20             cache: false,
21             success: function(data){
22                 data = JSON.parse(data);
23                 if(data.status){
24                     $("#pdf_viewer").attr('src', "<?=base_url('../assets/pdfjs/web/viewer.html?file='.$user-&gt;selected_assignment['id'].'/nulltrue')?>");
25                     $("#pdf_viewer").show();
26                 }
27             },
28             error: function (error){
29                 console.error(error);
30             },
31         });
32     </script>
33     <script src="=base_url('assets/ace/ace.js')?"></script>
34     <script type='text/javascript' src="=base_url("assets/js/shj_submit.js")?"></script>
35 <?= $this->endSection() ?>
36
37
38
39 <?= $this->section('main_content') ?>
40 <?php if ($error != 'none'): ?>
41 <p><?= $error ?></p>
42 <?php else: ?>
43     <p>Selected assignment: <span dir="auto"><?= $user->selected_assignment['name'] ?></span></p>
44     <p>Coefficient: <?= $coefficient ?>%</p>
45     <?= form_open_multipart('submit') ?>
46     <p class="input_p">
47         <label for="problems" class="tiny">Problem:</label>
48         <select id="problems" name="problem" class="sharif_input">
49             <option value="0" selected="selected">-- Select Problem --</option>
50             <?php foreach($problems as $problem): ?>
51                 <option dir="auto" value="= $problem['id']?&gt;"&gt;&lt;?= $problem['name'] ?&gt;&lt;/option&gt;
52             &lt;?php endforeach ?&gt;
53         &lt;/select&gt;
54         &lt;div class="shj_error"&gt;&lt;?= $validationError-&gt;hasError('problem') ? $validationError-&gt;getError('problem') : '' ?&gt;&lt;/div&gt;
55     &lt;/p&gt;
56     &lt;p class="input_p"&gt;
57         &lt;label for="languages" class="tiny"&gt;Language:&lt;/label&gt;
58         &lt;select id="languages" name="language" class="sharif_input"&gt;
59             &lt;option value="0" selected="selected"&gt;-- Select Language --&lt;/option&gt;
60             &lt;/select&gt;
61             &lt;!-- form_error('language', '&lt;div&gt;&lt;?= $language ?&gt;&lt;/div&gt;') ?&gt; --&gt;
62     &lt;/p&gt;
</pre

```

```

63  <details>
64    <summary>Submit by File Upload</summary>
65    <p class="input_p_upload_hidden">
66      <label for="file" class="tiny">File:</label>
67      <input type="file" id="file" class="sharif_input_medium" name="userfile" />
68    <?php if($upload_state == 'error'): ?>
69    <div class="shj_error">Error uploading file.</div>
70    <?php elseif ($upload_state == 'ok'): ?>
71    <div class="shj_ok">File uploaded successfully. See the result in 'All_Submissions'.</div>
72    <?php endif ?>
73  </p>
74  <p class="input_p_upload_hidden">
75    <input type="submit" value="Submit" class="sharif_input"/>
76  </p>
77 </details>
78 </form>
79
80 <br>
81 <details>
82   <summary>Code, Test, and Submit using built-in editor</summary>
83   <iframe id="pdf_viewer" src="" style="display:none"></iframe>
84   <div id="ide_wrap">
85     <fieldset id="editor_wrap">
86       <legend>Code</legend>
87       <div id="code_editor" ></div>
88     </fieldset>
89     <fieldset id="in_wrap">
90       <legend>Input</legend>
91       <textarea id="editor_input" class="in_out"></textarea>
92     </fieldset>
93     <fieldset id="out_wrap">
94       <legend>Output</legend>
95       <textarea id="editor_output" class="in_out" readonly></textarea>
96     </fieldset>
97   </div>
98   <br>
99   <button type="button" class="sharif_input" id="editor_save" disabled>Save</button>
100  <button type="button" class="sharif_input" id="editor_execute" disabled>Save & Execute</button>
101  <button type="button" class="sharif_input" id="editor_submit" disabled>Save & Submit</button>
102  <span id="ajax_status"></span>
103 </details>
104 <?php endif ?>
105 <?= $this->endSection() ?>

```

Kode C.32: base.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: base.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->renderSection('base')?>
7 <!DOCTYPE html>
8 <html lang="en">
9 <head>
10   <title><?= $this->renderSection('head_title') ?> - SharIF Judge</title>
11   <meta content="text/html" charset="UTF-8">
12   <link rel="icon" href="<?= _base_url('assets/images/favicon.ico') ?>" />
13   <link rel="stylesheet" type="text/css" href="<?= _base_url('assets/styles/main.css') ?>"/>
14   <script type="text/javascript" src="<?= _base_url('assets/js/jquery-1.10.2.min.js') ?>"></script>
15   <!-- You can use google's_cdn_for_loading_jquery:-->
16   <!--<script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>-->
17   <script type="text/javascript" src="<?= _base_url('assets/js/jquery-ui-1.10.3.custom.min.js') ?>"></script>
18   <link rel="stylesheet" href="<?= _base_url('assets/styles/flick/jquery-ui-1.10.3.custom.min.css') ?>"/>
19   <script type="text/javascript" src="<?= _base_url('assets/js/moment.min.js') ?>"></script>
20   <script type="text/javascript" src="<?= _base_url('assets/js/jquery.hoverIntent.minified.js') ?>"></script>
21   <script type="text/javascript" src="<?= _base_url('assets/js/jquery.cookie.js') ?>"></script>
22   <script type="text/javascript" src="<?= _base_url('assets/js/jquery.ba-resize.min.js') ?>"></script>
23
24   <script type="text/javascript" src="<?= _base_url('assets/noty/jquery.noty.js') ?>"></script>
25   <script type="text/javascript" src="<?= _base_url('assets/noty/layouts/center.js') ?>"></script>
26   <script type="text/javascript" src="<?= _base_url('assets/noty/layouts/bottomRight.js') ?>"></script>
27   <script type="text/javascript" src="<?= _base_url('assets/noty/themes/default.js') ?>"></script>
28
29   <link rel='stylesheet' type='text/css' href='<?= base_url('assets/nano_scroller/nanoscroller.css') ?>' />
30   <script type='text/javascript' src="<?= _base_url('assets/nano_scroller/jquery.nanoscroller.min.js') ?>"></script>
31
32   <link rel='stylesheet' type='text/css' href='<?= base_url('assets/font/fontawesome/css/fontawesome.min.css') ?>' />
33
34 </head>
35 <script>
36 shj={};
37 shj.site_url='<?= rtrim(site_url(), '/') ?>/';
38 shj.base_url='<?= rtrim(base_url(), '/') ?>/';
39 shj.csrf_token=$_cookie['shjcsrf'];
40 shj.offset=_moment('<?= shj_now.str() ?>').diff(moment());
41 shj.time=_moment();
42 shj.finish_time=_moment("<?= $finish_time ?>");
43 shj.extra_time=_moment.duration("<?= $extra_time ?>","seconds");
44 //notifications
45 shj.check_for_notifications=_false;
46 shj.notif_check_delay_=30;
47 shj.color_scheme_='github';
48 </script>
49
50 <script type="text/javascript" src="<?= _base_url('assets/js/shj_functions.js') ?>"></script>
51
52 <?= $this->renderSection('other_assets')?>

```

```

53| <body_id="body">
54| <a href="#page_title" class="skip_to_content">Skip to content</a>
55|
56| <?=_$this->include('templates/top_bar')_?>
57| <?=_$this->include('templates/side_bar')_?>
58| <div_id="main_container" class="scroll-wrapper">
59|   <div class="scroll-content">
60|     <div_id="page_title">
61|       <i class="fa_<?=_$this->renderSection('icon')_?>"></i>
62|       <h1_dir="auto"><?=_$this->renderSection('title')_?></h1>
63|       <?=_$this->renderSection('title_menu')?>
64|     </div>
65|     <div_id="main_content">
66|       <?=_$this->renderSection('main_content')?>
67|     </div>
68|   </div>
69| </div>
70| <?=_$this->renderSection('body_end')_?>
71| </body>
72| </html>
73|
74|
75|

```

Kode C.33: side_bar.php

```

1<!-- {#
2# SharIF Judge
3# file: side_bar.twig
4# author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5#} -->
6<?=_$this->renderSection('side_bar') ?>
7<div id="side_bar" class="sidebar_open">
8  <ul>
9    <li class="color-dashboard <?=_($selected=='_dashboard')_?_ '_selected'_: '_?>">
10      <a href=<?=_site_url('dashboard')_?>" aria-labelledby="dashboard-label">
11        <i class="fa_fa-dashboard_fa-lg"></i>
12        <span class="sidebar_text" id="dashboard-label">Dashboard</span>
13      </a>
14    </li>
15    <?php if ($user->level == 3): ?>
16    <li class="color-settings <?=_($selected=='settings')_?_ '_selected'_: '_?>">
17      <a href=<?=_site_url('settings')_?>" aria-labelledby="settings-label">
18        <i class="fa_fa-gear_fa-lg"></i>
19        <span class="sidebar_text" id="settings-label">Settings</span>
20      </a>
21    </li>
22    <li class="color-users <?=_($selected=='users')_?_ '_selected'_: '_?>">
23      <a href=<?=_site_url('users')_?>" aria-labelledby="users-label">
24        <i class="fa_fa-users_fa-lg"></i>
25        <span class="sidebar_text" id="users-label">Users</span>
26      </a>
27    </li>
28    <?php endif ?>
29    <li class="color-notifications <?=_($selected=='notifications')_?_ '_selected'_: '_?>">
30      <a href=<?=_site_url('notifications')_?>" aria-labelledby="notifications-label">
31        <i class="fa_fa-bell_fa-lg"></i>
32        <span class="sidebar_text" id="notifications-label">Notifications</span>
33      </a>
34    </li>
35    <li class="color-assignments <?=_($selected=='assignments')_?_ '_selected'_: '_?>">
36      <a href=<?=_site_url('assignments')_?>" aria-labelledby="assignments-label">
37        <i class="fa_fa-folder-open_fa-lg"></i>
38        <span class="sidebar_text" id="assignments-label">Assignments</span>
39      </a>
40    </li>
41    <li class="color-problems <?=_($selected=='problems')_?_ '_selected'_: '_?>">
42      <a href=<?=_site_url('problems')_?>" aria-labelledby="problems-label">
43        <i class="fa_fa-puzzle-piece_fa-lg"></i>
44        <span class="sidebar_text" id="problems-label">Problems</span>
45      </a>
46    </li>
47    <li class="color-submit <?=_($selected=='submit')_?_ '_selected'_: '_?>">
48      <a href=<?=_site_url('submit')_?>" aria-labelledby="submit-label">
49        <i class="fa_fa-location-arrow_fa-lg"></i>
50        <span class="sidebar_text" id="submit-label">Submit</span>
51      </a>
52    </li>
53    <li class="color-final_submissions <?=_($selected=='final_submissions')_?_ '_selected'_: '_?>">
54      <a href=<?=_site_url('submissions/final')_?>" aria-labelledby="final-submission-label">
55        <i class="fa_fa-map-marker_fa-lg"></i>
56        <span class="sidebar_text" id="final-submission-label">Final Submissions</span>
57      </a>
58    </li>
59    <li class="color-all_submissions <?=_($selected=='all_submissions')_?_ '_selected'_: '_?>">
60      <a href=<?=_site_url('submissions/all')_?>" aria-labelledby="all-submission-label">
61        <i class="fa_fa-bars_fa-lg"></i>
62        <span class="sidebar_text" id="all-submission-label">All Submissions</span>
63      </a>
64    </li>
65    <li class="color-scoreboard <?=_($selected=='scoreboard')_?_ '_selected'_: '_?>">
66      <a href=<?=_site_url('scoreboard')_?>" aria-labelledby="scoreboard-label">
67        <i class="fa_fa-star_fa-lg"></i>
68        <span class="sidebar_text" id="scoreboard-label">Scoreboard</span>
69      </a>
70    </li>
71    <li class="color-halloffame <?=_($selected=='halloffame')_?_ '_selected'_: '_?>">
72      <a href=<?=_site_url('halloffame')_?>" aria-labelledby="hall-of-fame-label">
```

```

73         <i class="fa_fa-list-alt_fa-lg"></i>
74         <span class="sidebar_text" id="hall-of-fame-label">Hall of Fame</span>
75     </a>
76   </li>
77 <?php if ($user->level == 3): ?>
78 <li class="color-logs<?=$selected=='logs'?>'>
79   <a href="=$site_url('logs')?&gt;" aria-labelledby="24-hour-log-label"&gt;
80     &lt;i class="fa_fa-book_fa-lg"&gt;&lt;/i&gt;
81     &lt;span class="sidebar_text" id="24-hour-log-label"&gt;24-hour Log&lt;/span&gt;
82   &lt;/a&gt;
83 &lt;/li&gt;
84 &lt;?php endif ?&gt;
85 &lt;/ul&gt;
86 &lt;div id="sidebar_bottom"&gt;
87   &lt;p&gt;
88     &lt;a href="https://github.com/ifunpar/ShariF-Judge" target="_blank"&gt;&amp;copy; SharIF Judge &lt;?= SHJ_VERSION ?&gt;&lt;/a&gt;
89     &lt;a href="https://github.com/ifunpar/ShariF-Judge/tree/docs" target="_blank"&gt;Docs&lt;/a&gt;
90   &lt;/p&gt;
91   &lt;p class="timer"&gt;&lt;/p&gt;
92   &lt;div id="shjCollapse" class="pointer"&gt;
93     &lt;i id="collapse" class="fa_fa-caret-square-o-left_fa-lg"&gt;&lt;/i&gt;&lt;span class="sidebar_text"&gt;Collapse Menu&lt;/span&gt;
94   &lt;/div&gt;
95 &lt;/div&gt;
96 &lt;/div&gt;
</pre

```

Kode C.34: simple_header.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: simple_header.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->renderSection('simple_header') ?>
7 <head>
8   <title><?= $title ?> - SharIF Judge</title>
9   <meta content="text/html" charset="UTF-8">
10  <link rel="icon" href="=$base_url("/assets/images/favicon.ico")?"/&gt;"
11  <link rel="stylesheet" type="text/css" href="=$base_url("/assets/styles/login.css")?"/&gt;"/
12 </head>

```

Kode C.35: top_bar.php

```

1 <!-- {#
2 # SharIF Judge
3 # file: top_bar.twig
4 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
5 #} -->
6 <?= $this->renderSection('top_bar') ?>
7 <div id="top_bar" class="color-=$selected?&gt;"&gt;
8   &lt;div class="top_object_shj_menu" id="user_top"&gt;
9     &lt;a href="<?=$site_url('profile')?&gt;" id="profile_link" aria-label="Profile"&gt;&lt;i class="fa_fa-user"&gt;&lt;/i&gt;&lt;/a&gt;
10    &lt;div class="top_menu_user-menu"&gt;
11      &lt;div class="gravatar"&gt;&lt;img src="https://www.gravatar.com/avatar/&lt;?=$md5($user-&gt;email)?&gt;?s=70&amp;d=identicon" /&gt;&lt;/div&gt;
12      &lt;div class="name"&gt;&lt;i class="fa_fa-user"&gt;&lt;/i&gt; &lt;?= $user-&gt;username ?&gt;&lt;/div&gt;
13      &lt;div class="menu-controls"&gt;
14        &lt;a href="<?=$site_url('/logout')?&gt;" class="logout-btn"&gt;&lt;i class="fa_fa-sign-out"&gt;&lt;/i&gt; Sign Out&lt;/a&gt;
15        &lt;a href="<?=$site_url('/profile')?&gt;" class="profile-btn"&gt;&lt;i class="fa_fa-wrench"&gt;&lt;/i&gt; Profile&lt;/a&gt;
16      &lt;/div&gt;
17    &lt;/div&gt;
18  &lt;/div&gt;
19  &lt;div class="top_object_countdown" id="countdown"&gt;
20    &lt;div class="time_block"&gt;
21      &lt;span id="time_days" class="countdown_num"&gt;&lt;/span&gt;&lt;br&gt;
22      &lt;span class="time_label" id="days_label"&gt;&lt;/span&gt;
23    &lt;/div&gt;
24    &lt;div class="time_block"&gt;
25      &lt;span id="time_hours" class="countdown_num"&gt;&lt;/span&gt;&lt;br&gt;
26      &lt;span class="time_label" id="hours_label"&gt;&lt;/span&gt;
27    &lt;/div&gt;
28    &lt;div class="time_block"&gt;
29      &lt;span id="time_minutes" class="countdown_num"&gt;&lt;/span&gt;&lt;br&gt;
30      &lt;span class="time_label" id="minutes_label"&gt;&lt;/span&gt;
31    &lt;/div&gt;
32    &lt;div class="time_block"&gt;
33      &lt;span id="time_seconds" class="countdown_num"&gt;&lt;/span&gt;&lt;br&gt;
34      &lt;span class="time_label" id="seconds_label"&gt;&lt;/span&gt;
35    &lt;/div&gt;
36  &lt;/div&gt;
37  &lt;div class="top_object_countdown" id="extra_time"&gt;
38    &lt;i class="fa_fa-plus-square_fa-2x"&gt;&lt;/i&gt;
39    &lt;div class="time_block"&gt;
40      &lt;span&gt;Extra&lt;/span&gt;&lt;br&gt;&lt;span&gt;Time&lt;/span&gt;
41    &lt;/div&gt;
42  &lt;/div&gt;
43  &lt;div id="shj_logo" class="top_left"&gt;
44    &lt;a href="<?=$site_url('/')?&gt;"&gt;
45      &lt;img src="<?=$base_url('/assets/images/logo_small.png')?"/&gt;" aria-label="Logo_ShariF_Judge"/&gt;
46      &lt;h1 class="shjlogo-text"&gt;SharIF &lt;span&gt;Judge&lt;/span&gt;&lt;/h1&gt;
47    &lt;/a&gt;
48  &lt;/div&gt;
49 &lt;?php if ($user-&gt;level &gt;= 2): ?&gt;
50 &lt;div class="top_object_shj_menu_top_left" tabindex="0" id="admin_tools_top"&gt;
51   Tools
52   &lt;ul class="top_menu"&gt;
53     &lt;li&gt;&lt;a href="<?=$site_url('rejudge')?&gt;"&gt;Rejudge&lt;/a&gt;&lt;/li&gt;
</pre

```

```
54         <li><a href="=site_url('queue')?&gt;"Submission Queue</a></li>
55         <li><a href="=site_url('moss/'.$user-&gt;selected_assignment['id'])?&gt;"Cheat Detection</a></li>
56     </ul>
57 </div>
58 <?php endif ?>
59 <div class="top_object_shj_menu_top_left" id="select_assignment_top">
60     <a href="=site_url('assignments')?&gt;"&gt;&lt;span dir="auto" class="assignment_name"<?= strlen($user->selected_assignment['name']) > 30 ? array_slice($user->selected_assignment['name'],0,30) . '...' : $user->selected_assignment['name'] ?></span></a>
61     <ul class="top_menu" id="select_assignment_menu">
62         <?php $reversed_assignment = array_slice(array_reverse($all_assignments), 0, 5); ?>
63         <?php foreach ($reversed_assignment as $assignment_item): ?>
64             <li class="assignment_block_select_assignment" tabindex="0">
65                 <i class="fa_<?= $assignment_item['id'] == $user->selected_assignment['id'] ? 'fa-check-square-o_color6' : 'fa-square-o'_?>" data-id="= $assignment_item['id']?&gt;"&gt;&lt;/i&gt;
66                 &lt;span class="assignment_item" dir="auto"&gt;&lt;?= $assignment_item['name'] ?&gt;&lt;/span&gt;
67             &lt;/li&gt;
68         &lt;?php endforeach; ?&gt;
69         &lt;li&gt;
70             &lt;a href="<?=site_url('assignments')?&gt;"All Assignments</a>
71         </li>
72     </ul>
73 </div>
74 <div class="top_object_top_left_shj-spinner" style="display:none;">
75     <i class="fa_fa-refresh_fa-spin_fa-lg"></i>
76 </div>
77 </div>
```