

Introduksi:	2
Developer:	2
Running web app	2
Errors:	3
Database	3
Config	4
env	4
App.php	5
Database.php	5
Email.php	6
Filters.php & Routes.php	7
Validation.php	8
User:	8
Home page	8
Dashboard	9
Profile	11
Schedule	11
Manage Users and Permissions	12
Let's manage roles	12
Role permissions	13
Host requests	14
View Reports	15
Classes	16
Create Classes	16
Explore Classes	17
View Class	18
Edit Class	19
Meeting	20
Notes	20
Add, Edit & View Note	20
People	21
Invite User & View people	21
Tests	22
Tests page	22
Edit Test & Edit Question	23
Begin Test	24
Take Test & Submit Exam	25
Tools:	27

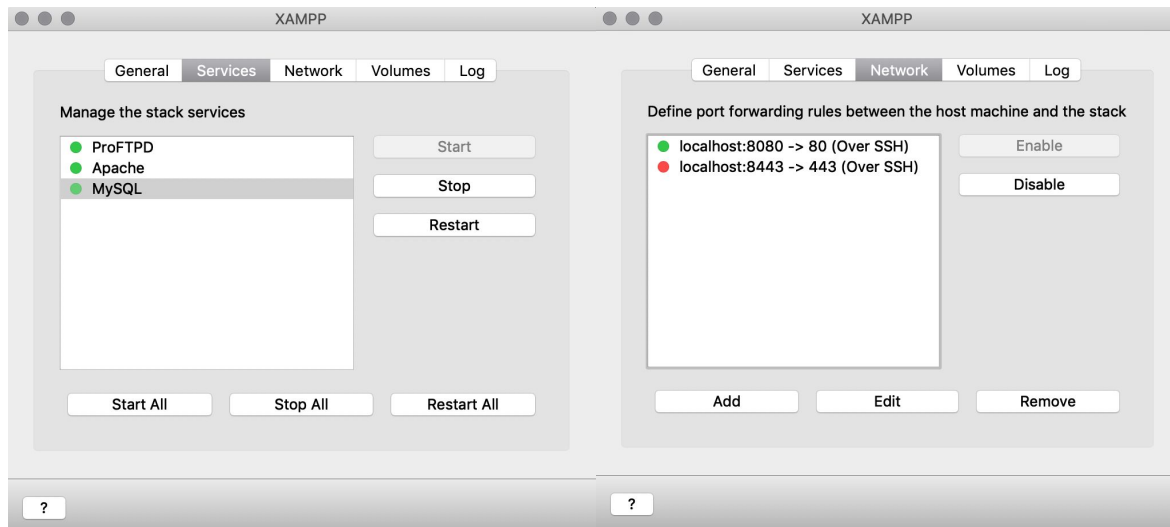
Introduksi:

Projek ini adalah aplikasi berbasis web dengan tampilan mobile untuk sistem penjadwalan kelas online. Aplikasi ini bisa digunakan untuk Sign up, Login, Create Class, Edit Class, Remove Class, Create Meeting, Invite People, Add Meeting Note, View Schedule, Pre Test dan Post Test sesuai deskripsi yang diberikan. Aplikasi dibuat dengan PHP 7.4.7 dengan Framework CodeIgniter 4, Database MySQL, Frontend CSS Framework dengan Bootstrap 4 dan JavaScript dan module Fullcalendar untuk display jadwal. User Manual ini dapat dibagi menjadi beberapa bagian. Bagian Developer bisa dirujuk saat menambah fitur dan files mana yang telah diganti atau saat set up web app di server diri sendiri. Bagian User bisa digunakan oleh User dan Developer keduanya. Bagian ini memberikan petunjuk cara menggunakan fitur untuk tipe role berbeda dan juga apa yang terjadi di komponen backend appnya dan file yang terlibat. Info Backend memberikan penjelasan singkat tentang apa yang terjadi didalam database dan kolom/variabel yang di update oleh setiap aksi user. Jika terjadi error, manual ini bisa digunakan untuk mencari Controller, Model dan View files yang mengandung fungsi yang terlibat.

Developer:

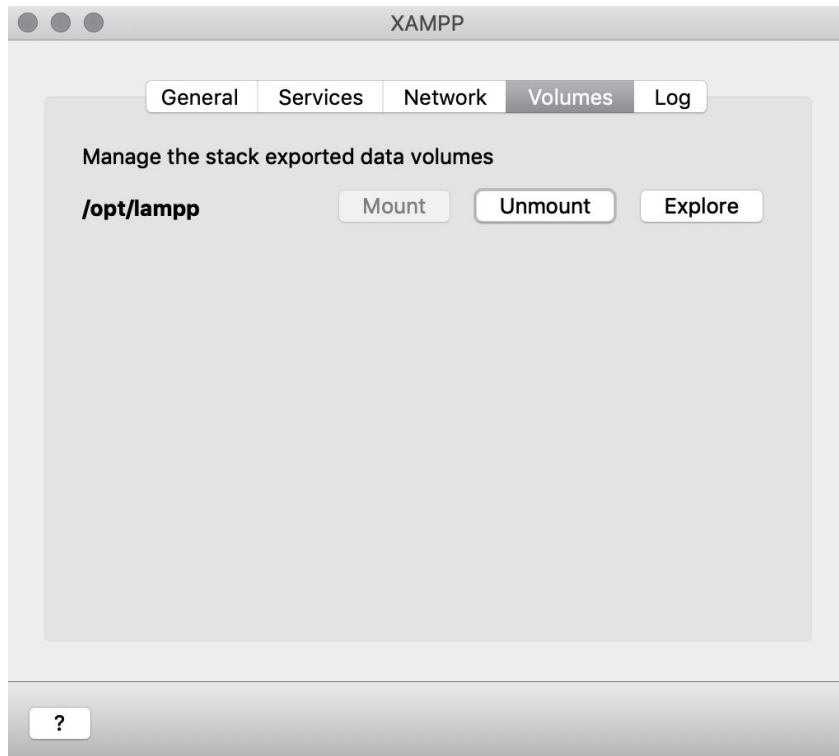
Running web app

Saya menggunakan XAMPP dan menyalakan module proFTPD, Apache dan MySQL stack services melalui tab 'Services' dengan tombol 'Start All'. Kemudian, localhost bisa di akses menggunakan port 8080 setelah di 'Enable' melalui tab 'Network'.



Setelah localhost di set up, folder web app **K24** yang di clone dari Github dimasukan ke folder /lampp/htdocs/ setelah /opt/lampp di 'Mount'.

(<https://github.com/rachael-soh/Online-Learning-webapp.git>)



Akhirnya, web app bisa langsung diakses dari link: <http://localhost:8080/k24/public/>

Errors:

Awalnya saya sering kena error.

```
CodeIgniter\Cache\Exceptions\CacheException  
Cache unable to write to /htdocs/k24/writable/cache/
```

Tetapi saya menggunakan command line command (ganti path sesuai dengan path diri sendiri):

```
chmod 777 -R /htdocs/k24/writable
```

Info: <https://forum.codeigniter.com/thread-73000.html>

Database

File: ~/htdocs/k24/k24 (1).sql

Database bisa langsung diimport ke PHPMyAdmin page (<http://localhost:8080/phpmyadmin/>) atau developer bisa membuat database baru. Web app ini mempunyai 13 tabel mySQL.

- **users:** *user_id* (Primary Key) , *fname*, *lname*, *email*, *password* (hashed), *role_id* (1=Admin, 2= Host, 3= Peserta), *created_at* (tanggal), *updated_at* (tanggal), *user_status* (0 = tidak aktif, 1=aktif)
- **user_classes:** *uc_id* (Primary key), *user_id*, *class_id*, *host* (0= bukan host, 1=host kelas)
- **classes:** *class_id* (Primary key), *class_name*, *description*, *call_link* (URL call), *start_time*, *end_time*, *start_date*, *end_date*, *recurring* (kelas berulang setiap: 1= 1 hari,

2= setiap minggu, 3= setiap bulan), *dow* (hari dalam seminggu: NULL = kelas hanya terjadi pada 1 hari, 1=senin, 2=selasa, 3= rabu, 4=kamis, 5=jumat, 6=sabtu, 7=minggu), *pretest_id* (id sesuai dengan tes di tabel tests), *posttest_id* (id sesuai dengan tes di tabel tests), *class_status* (0 = belum mulai, masih bisa join, 1 = kelas sudah dimulai, tidak bisa join lagi, 2 = kelas sudah selesai dan tidak aktif), *color* (warna random)

- **notes:** *note_id* (Primary key), *class_id*, *note_name*, *note_doc*, *note_path* (nama note setelah diberi nama CI)
- **tests:** *test_id* (Primary key), *class_id*, *total_score* (total nomor soal), *test_date* (NULL kalau pretest), *start_time* (NULL kalau pretest), *end_time* (NULL kalau pretest), *duration* (durasi dalam menit), *test_status* (1=pretest, bisa dilakukan tanpa batas, 2 = posttest, hanya bisa dilakukan sekali, 3=kelas sudah selesai, tes tidak aktif),
- **questions:** *question_id* (Primary key), *test_id* (soal ini untuk test yang mana), *question*
- **options:** *option_id* (Primary key), *question_id*, *option_desc*, *ans* (0=bukan jawaban, 1=jawaban memberi 1 point).
- **user_test_answer:** *uta_id* (Primary key), *user_id*, *test_id*, *question_id*, *marks* (0/1 poin).
- **user_scores:** *us_id* (Primary key), *user_id*, *class_id*, *test_id*, *score* (nilai sebagai persentase), *test_status* (1=pretest, 2=posttest)
- **roles:** *role_id* (Primary key), *role_desc*
- **perms:** *perm_id* (Primary key), *perm_name*
- **role_perm:** *role_perm_id* (Primary key), *role_id*, *perm_id*
- **hostRequest:** *request_id* (Primary key), *user_id*, *class_id*, *request_status* (0= tidak aktif/sudah di manage admin, 1=pending request)

Config

Files:

~/htdocs/k24/app/Config/

- App.php
- Database.php
- Email.php
- Filters.php
- Routes.php
- Validation.php

~/htdocs/k24/

- Env

env

Saya hanya mengganti CI_Environment di file ini menjadi 'development' agar keluar error message yang detail. Tetapi, setting ini bisa diganti balik ke 'production' agar error message tidak muncul. Database dan App settings disini juga bisa diganti sesuai dengan info yang diisi di App.php dan Database.php.

App.php

Di App.php, developer bisa mengganti *baseURL* webapp. Untuk sementara, *baseURL* si set sebagai berikut: '<http://localhost:8080/k24/public/>'. Tetapi, ini akan berubah tergantung dengan port yang digunakan atau lokasi folder k24. *baseURL* harus diubah sesuai link terbaru agar function yang menggunakan variabel ini tetap berfungsi.

```
/*
|-----
| Base Site URL
|-----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|   http://example.com/
|
| If this is not set then CodeIgniter will try guess the protocol, domain
| and path to your installation. However, you should always configure this
| explicitly and never rely on auto-guessing, especially in production
| environments.
|
*/
*/
public $baseURL = 'http://localhost:8080/k24/public/';
/*
```

Jangan lupa mengganti Timezone jika ingin digunakan di time zone berbeda.

```
/*
|-----
| Application Timezone
|-----
|
| The default timezone that will be used in your application to display
| dates with the date helper, and can be retrieved through app_timezone()
|
*/
*/
public $appTimezone = 'Asia/Jakarta';
/*
```

Database.php

File ini perlu diganti agar bisa mengakses database yang akan digunakan web app ini. Info database mySQL perlu diisi di variabel *\$default* . Info ini bisa ditemukan di PHPMYAdmin saat menggunakan XAMPP (<http://localhost:8080/phpmyadmin/>). Database perlu dibuat dulu dan username, password dan nama database dimasukkan 'username', 'password' dan 'database'. 'hostname' dan 'port' juga diisi sesuai *baseURL*.

```

/**
 * The default database connection.
 *
 * @var array
 */
public $default = [
    'DSN'      => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'k24',
    'DBDriver' => 'MySQLi',
    'DBPrefix' => '',
    'pConnect' => false,
    'DBDebug'  => (ENVIRONMENT !== 'production'),
    'cacheOn'  => false,
    'cacheDir' => '',
    'charset'  => 'utf8',
    'DBCollat' => 'utf8_general_ci',
    'swapPre'  => '',
    'encrypt'  => false,
    'compress' => false,
    'strictOn' => false,
    'failover' => [],
    'port'     => 8080,
];

```

Email.php

Image berikutnya adalah variabel yang telah diganti dari file original 'Email.php' CodeIgniter agar web app bisa mengirim email. *\$protocol*, *\$mailPath*, *\$SMTPHost*, *\$SMTPCrypto*, *\$mailType* telah di set sesuai dengan private function `sendEmail` di File `Controller/Classes.php` saat invite user ke kelas. Untuk sementara, function ini tidak berfungsi karena *\$SMTPUser* dan *\$SMTPPass* belum diisi dengan email dan password pengirim invite.

```

/**
 * The mail sending protocol: mail, sendmail, smtp
 *
 * @var string
 */
public $protocol = 'smtp';

/**
 * The server path to Sendmail.
 *
 * @var string
 */
public $mailPath = '/usr/sbin/sendmail';

/**
 * SMTP Server Address
 *
 * @var string
 */
public $SMTPHost = 'smtp.gmail.com';

```

```

/**
 * SMTP Username
 *
 * @var string
 */
public $SMTPUser = '';

/**
 * SMTP Password
 *
 * @var string
 */
public $SMTPPass = '';

/**
 * SMTP Port
 *
 * @var integer
 */
public $SMTPPort = 465;

```

```

/**
 * SMTP Encryption. Either tls or ssl
 *
 * @var string
 */
public $SMTPCrypto = 'ssl';

```

```

/**
 * Type of mail, either 'text' or 'html'
 *
 * @var string
 */
public $mailType = 'html';

```

Filters.php & Routes.php

Untuk sementara, web app ini hanya ada 2 filter: Auth.php dan noAuth.php. Filter ini hanya cek jika user logged in atau tidak, kalau user sudah tidak logged in, ia akan dibawa balik ke log in page. Kalau ada filter lain yang ingin ditambah, file dibuat di folder \App\Filters\ dan langsung ditambah ke list *\$aliases*.

```

// Makes reading things below nicer,
// and simpler to change out script that's used.
public $aliases = [
    'csrf'      => \CodeIgniter\Filters\CSRF::class,
    'toolbar'    => \CodeIgniter\Filters\DebugToolbar::class,
    'honeypot'   => \CodeIgniter\Filters\Honeypot::class,
    'auth'       => \App\Filters\Auth::class,
    'noauth'     => \App\Filters\Noauth::class,
];

```

Setelah itu, filternya bisa di apply kepada route berbeda seperti berikut. Route juga bisa ditambah jika menambah view atau controller baru.

```

/**
 * -----
 * Route Definitions
 * -----
 */

// We get a performance increase by specifying the default
// route since we don't have to scan directories.
$routes->get('/', 'Pages::index', ['filter' => 'noauth']);
$routes->get('logout', 'Pages::logout');
$routes->match(['get','post'],'signup', 'Pages::signup', ['filter' => 'noauth']);
$routes->match(['get','post'],'profile', 'Pages::profile', ['filter' => 'auth']);
$routes->get('dashboard', 'Dashboard::index', ['filter' => 'auth']);
$routes->get('manageUsers', 'Dashboard::manageUsers', ['filter' => 'auth']);
$routes->get('viewReports', 'Dashboard::viewReports', ['filter' => 'auth']);
$routes->get('getUsers', 'ManageUsers::getUsers', ['filter' => 'auth']);

```

Validation.php

Jika ingin menambah 'Rules' baru, file harus dibuat di folder `\App\Validation\` dan path ditambah ke list `$ruleSets` seperti `UserRules` dan `ClassRule`.

```

/**
 * Stores the classes that contain the
 * rules that are available.
 *
 * @var array
 */
public $ruleSets = [
    \CodeIgniter\Validation\Rules::class,
    \CodeIgniter\Validation\FormatRules::class,
    \CodeIgniter\Validation\FileRules::class,
    \CodeIgniter\Validation\CreditCardRules::class,
    \App\Validation\UserRules::class,
    \App\Validation\ClassRules::class,
];

```

User:

Home page

Bisa dilihat: Admin, Host, Peserta

Link: <http://localhost:8080/k24/public/>

Saat masuk link tersebut, user akan langsung dibawa ke login page. User bisa langsung log in menggunakan email dan password akun mereka. Jika tidak ada akun, user bisa ke sign up page melalui navbar atau link *'Don't have an account?'* dan membuat akun baru. Setelah user

membuat akun, aler sukses akan muncul dan user bisa langsung log in akun barunya melalui log in page.

Backend: User baru akan ditambah ke database 'users'. Setelah user berhasil log in, user session akan diset agar user_id, role user dan nama user bisa di akses. Variabel *isLoggedIn* juga di set jadi *true* agar user bisa dihentikan melihat page yang tidak bisa diakses sebelum log in.

Files: Controller: Pages.php, Models: UserModel.php, View: start-header.php, login.php, signup.php, footer.php

The screenshot shows a web interface for 'APOTEK K24' with the tagline 'Sobat Sehat Kita-Kita'. It features two main sections: a login form on the left and a signup form on the right. The login form includes fields for Email (pre-filled with 'rsoh@g.hmc.edu') and Password (masked with dots), with a 'Login' button and a link 'Don't have an account?'. The signup form includes fields for First Name, Last Name, Email (pre-filled with 'rsoh@g.hmc.edu'), Password, and Confirm Password, with a 'Sign up' button and a link 'Already have an account?'. Both sections have a 'Login' or 'Sign up' button at the top. The interface is clean with a green and white color scheme.

Dashboard

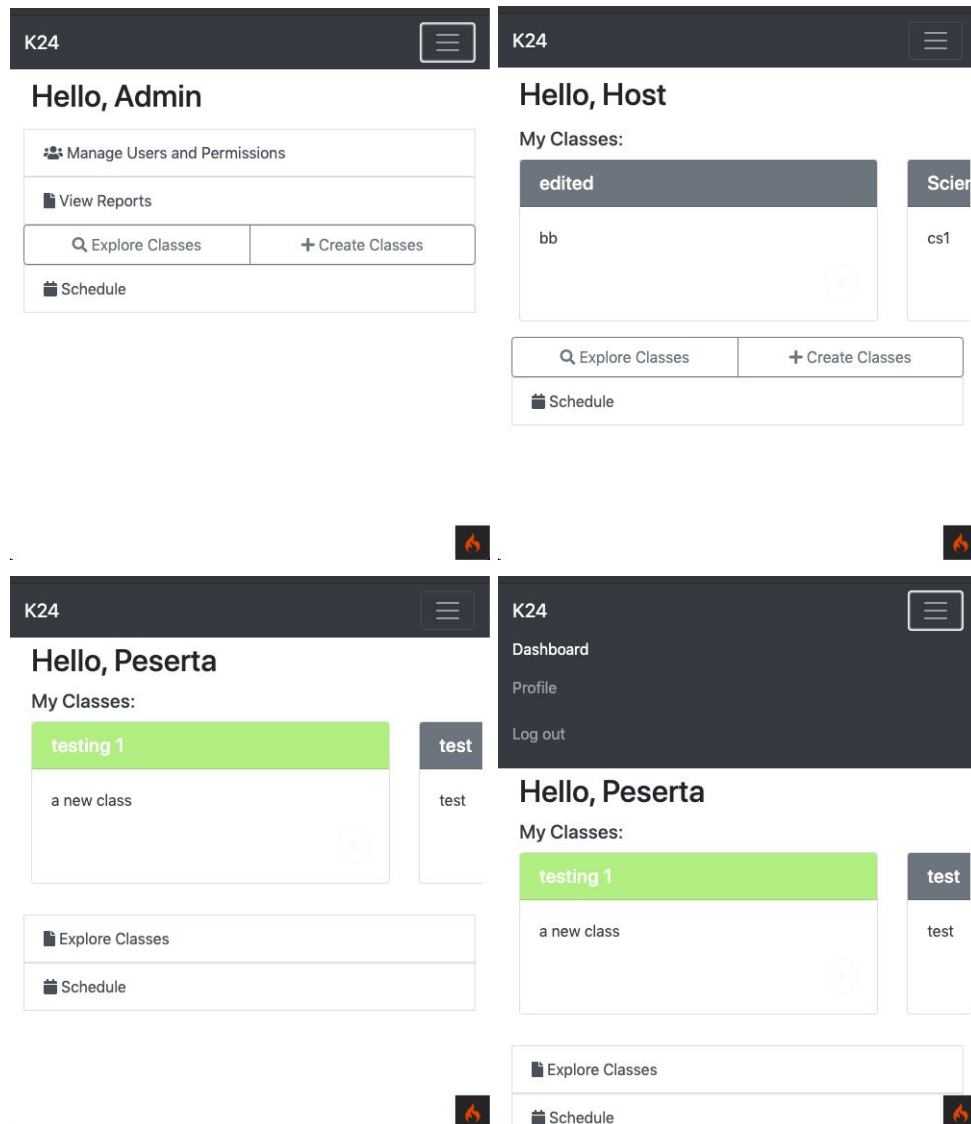
Bisa dilihat: Admin, Host, Peserta

Setelah user log in, ia langsung dibawa ke dashboard. Saat user membuat account, ia langsung diberikan role 'Peserta'. Tetapi user bisa diberikan role 'Admin', 'Host' atau 'Peserta'. Role masing-masing akan mempunyai dashboard berbeda karena mereka mempunyai permission berbeda. Admin akan mempunyai opsi 'Manage Role and Permission' dan 'View Reports' karena hanya Admin bisa mengganti role dan permisi user dan melihat nilai test semua user. Navbar semua role mempunyai opsi 'Profile' untuk mengganti password dan opsi 'Log out' untuk log out dari account. Host dan Peserta juga mempunyai sliding list yang memperlihatkan kelas yang telah terdaftar atau kelas yang di host oleh mereka untuk akses lebih mudah. Jika panel kelasnya berwarna abu, kelas itu sudah berakhir, tetapi jika masih berwarna, kelas itu sedang berjalan. Host dan Admin juga mempunyai opsi 'Create Classes' untuk membuat kelas. Selain itu, semua role mempunyai opsi 'Explore classes' untuk melihat semua kelas dan 'Schedule' agar bisa melihat jadwal kelas.

Backend: saat load dashboard, role user akan di cek dan memperlihatkan dashboard tertentu. Jika Host atau Peserta, info kelas akan ditarik dari table 'users', 'user_classes' dan 'classes'. Tanggal kelas akan dicek dan akan diperlihatkan sesuai status aktifnya. Jika kelas sudah tidak

aktif, 'class_status' akan di update di tabel 'classes' dan nilai tes akan di update sesuai status kelas.

Files: Controller: Dashboard.php, Models: ClassModel.php, Views: header.php, footer.php, fullCalendar.php, exploreClasses.php, createClass.php, viewReports.php, manageUsers.php, dashboard.php



Profile

Bisa dilihat: Admin, Host, Peserta

Profile page bisa digunakan untuk mengganti nama user dan password, tetapi email address tidak bisa diganti. Setelah info diganti, user perlu menekan tombol update untuk save data baru.

Backend: user info di update ke table 'users'

Files: Controller: Pages.php, Models: UserModel.php, Views: dashboard-header.php, footer.php, profile.php

Peserta Peserta

First Name

Last Name

Email address

Password

Confirm Password



Schedule

Bisa dilihat: Admin, Host, Peserta

Schedule akan memperlihatkan jadwal kelas user tersebut. Jika user adalah Admin, schedule akan memperlihatkan semua kelas. Tetapi, jika user adalah Host atau Peserta, schedule akan memperlihatkan kelas yang ia telah daftarkan atau adalah hostnya. User bisa memilih view “month” dan melihat jadwal secara kalender atau view “list” yang memperlihatkan daftar kelas pada hari itu.

Backend: info semua kelas diambil dari tabel ‘classes’ untuk Admin dan kelas user tersebut diambil dari table ‘classes’ dan ‘user_classes’ untuk Host dan Peserta. Info kelas user akan dibuat JSON encoding. Kemudian, JSON encoding yang terbuat akan dimasukkan module fullcalendar dan semua kelas dibuat event masing-masing dan diberi display seperti tersebut.

Files: Controller: Dashboard.php, Models: ClassModel.php, Views: fullCalendar.php

My Schedule

August 2020

month list

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	10:4 Math	10:3 abc	7: Scienc		10:3 abc 11:5 Engli 3:2 bikin	
16	17	18	19	20	21	22
11:4 color 11:4 color 11:4 color	11:5 Engli				2:2 testir	
23	24	25	26	27	28	29
30	31	1	2	3	4	5

today



My Schedule

Aug 16 – 22, 2020

month list

Sunday	August 16, 2020
11:42am - 12:42pm	• colored
11:43am - 12:43pm	• color test
11:52am - 12:52pm	• color again
Monday	August 17, 2020
11:50am - 12:50pm	• English 1
Friday	August 21, 2020
2:21pm - 3:21pm	• testing 1

today



Manage Users and Permissions

Bisa dilihat: Admin

Jika Admin masuk ke page “Manage Users and Permissions”, ia akan dibawa ke page ini.

Files: Controller: Dashboard.php, Models: ClassModel.php, Views: header.php, footer.php, manageUsers.php

K24	☰
Home > Manage User & Permission	
Let's manage roles	
Role permissions	
Host requests	



Let's manage roles

Bisa dilihat: Admin









Semua user bisa diganti rolenya menggunakan dropdown list dan di save dengan tombol biru.









Jika akun user ingin dihapus, Admin bisa menonaktifkan user dengan tombol merah. Jika akun

user sudah tidak aktif, ia akan berwarna abu dan akun bisa diaktifkan lagi oleh Admin. Jika admin juga bisa manage user berdasarkan role dengan tombol *All*, *Admin*, *Host* dan *Peserta*.

Backend: user role_id akan di update kalau role diganti. Jika user dinonaktifkan atau diaktifkan, user_status akan di update di tabel 'users'

Files: Controller: ManageUsers.php, Models: ManageUser.php, ManagePermission.php, Views: header.php, footer.php, manageRoles.php

Name	Role	Action
Peserta Peserta	Peserta	 
Rachael Soh	Admin	 
Host Host	Host	 
ruddy ruddy	Host	 

Name	Role	Action
Peserta Peserta	Peserta	 
Rachael Soh	Admin	 
Host Host	Host	 
ruddy ruddy	Host	 

Role permissions

Bisa dilihat: Admin

Permisi role berbeda bisa langsung diganti melalui page ini. Role bisa dipilih dan permisi role masing-masing bisa langsung dilihat. Untuk menambah dan menghapus permisi, Admin bisa memilih permisi yang ingin ditambah atau dihapus role itu dan langsung di save. Permisi ini akan diterapkan semua user yang mempunyai role itu! Jika role Peserta diberi permisi 'Create note' semua user yang mempunyai role Peserta bisa 'Create note'. Permisi tidak bisa di set user satu-satu.

Backend: tabel 'role_perm' akan ditambah atau dihapus.

Files: Controller: ManagePermissions.php, Models: ManagePermission.php, managePermission.php, Views: header.php, footer.php, managePermissions.php

K24

Home > Manage User & Permission > Role Permissions

Admin

Host

Peserta

☐ View all reports
☐ Manage Role
☐ Membuat class
☐ Mengundang peserta masuk class
☐ Request host
☐ Create note
☐ Edit note
☒ View note
☐ Delete note
☐ Create test
☐ Edit test

Peserta

☐ View all reports
☐ Manage Role
☐ Membuat class
☐ Mengundang peserta masuk class
☐ Request host
☐ Create note
☐ Edit note
☒ View note
☐ Delete note
☐ Create test
☐ Edit test
☒ Take test
☒ View test score
☒ Join class
☐ Remove class
☐ Edit Class
☐ Approve Host Request
☐ Remove Peserta from class

Host requests

Bisa dilihat: Admin

Admin bisa menyetujui atau menolak request Peserta menjadi Host sebuah kelas. Host bisa mengirimkan request ke Admin untuk membuat sebuah peserta menjadi host sebuah kelas. Jika peserta menjadi host kelas, ia akan diberikan permisi host untuk kelas itu saja. Ex. Jika request 'Peserta Peserta' untuk menjadi Host kelas 'testing 1' disetujui, ia akan bisa membuat note baru, melihat nilai test, etc. untuk kelas 'testing 1'.

Backend: kolom host akan diganti jadi 1 untuk user itu di tabel 'user_classes'. Host request akan di nonaktifkan di tabel 'hostRequest'.

Files: Controller: ManageUsers.php, Models: manageUser.php, managePermission.php, Views: header.php, footer.php, hostRequests.php

K24

Home > Manage User & Permission > Host Request

Name	Class Name	Action
Peserta Peserta	testing 1	<div>✓</div> <div>✗</div>

View Reports

Bisa dilihat: Admin



Admin bisa melihat semua nilai tes user. Nilai tes bisa dilihat melalui kelas atau langsung per user. Jika ingin melihat nilai tes per kelas, Admin bisa memilih opsi “By Class” dan memilih kelas yang ingin dilihat. Kemudian, Class Report akan memperlihatkan nilai tes semua peserta yang terdaftar kelas itu. Jika ingin melihat nilai tes user tertentu, Admin bisa memilih opsi “By User” dan melihat nilai tes user langsung. Search bar juga bisa digunakan untuk mencari kelas atau user tertentu, jika ingin mencari user melalui search bar, opsi ‘By user’ harus dipilih dulu. Setelah selesai menggunakan search bar, tombol list sebelah tombol search bisa dipilih untuk memperlihatkan semua user dan kelas lagi.

Backend: nilai tes akan ditarik dari tabel ‘user_scores’ dan diperlihatkan secara tabel. (Lihatlah Model `TestModel` `getClassReports` dan `getAllUserReports` untuk melihat SQL Query yang format nilai seperti yang diperlihatkan di page tersebut)

Files: Controller: `TestReport.php`, Models: `TestModel.php`, `ManagePermission.php`, `ClassModel.php`, `QuestionModel.php`. Views: `header.php`, `footer.php`, `viewReports.php`, `classReport.php`.

K24

Home > View Reports

By Class

By User

testing 1

a new class
21 August 2020 - 21 August 2020
14:21 - 15:21

edited

K24

Home > View all reports > Class Report

Name	Class	PreTest	PostTest	Progress	Pass
Peserta	testing	-	-	-	-
Peserta	1				

K24

Home > View Reports

Search

By Class

By User

Name	Class	PreTest	PostTest	Progress	Pass
Rachael Soh	edited	-	-	-	-
Host Host	edited	-	-	-	-
ruddy ruddy	edited	-	-	-	-
Peserta Peserta	test	100	0	Turun	Fail
Rachael	another	0	50	Maik	D

Classes

Create Classes

Bisa dilihat: Admin, Host

Admin dan Host bisa membuat kelas baru melalui tombol ‘Create classes’. Untuk membuat kelas baru, user perlu memberi nama kelas, deskripsi kelas, link video call untuk meeting kelas (google meet atau zoom link), waktu mulai, waktu berakhir dan tanggal kelas. Kelas bisa dibuat berulang (recurring), jika kelas hanya ‘1 Day’ user hanya perlu mengisi tanggal kelas. Tetapi, jika kelas terjadinya ‘Weekly’ atau ‘Monthly’, user perlu menentukan hari-hari kelas akan bertemu agar bisa dimasukkan jadwal kelas. Setelah info kelas diisi, kelas perlu di save dan user yang membuat kelas akan dijadikan host kelas ini.

Backend: kelas baru akan ditambah ke tabel ‘classes’ dan diberi status 0 karena masih bisa didaftar, diberikan warna random dan pretest_id dan posttest_id yang baru. Entry tes baru akan ditambah ke tabel ‘tests’ dengan pretest_id dan posttest_id yang baru dibuat oleh kelas ini. User yang membuat kelas ini akan ditambahkan ke tabel ‘user_classes’ dan kolom host diberikan angka 1.

Files: Controller: Classes.php, Models: ClassModel.php, ManagePermission.php. Views: header.php, footer.php, createClass.php

K24

Home > Create class

Create New Class

Class Name

Description

Call link

Start Time

--:--

End Time

--:--

Recurring

☒ 1 day
☐ Weekly
☐ Monthly

Start Date

mm/dd/yyyy

Create

Recurring

☐ 1 day
☒ Weekly
☐ Monthly

Start Date

mm/dd/yyyy

End Date

mm/dd/yyyy

Days

☐ Monday
☐ Tuesday
☐ Wednesday
☐ Thursday
☐ Friday
☐ Saturday
☐ Sunday

Create

Explore Classes

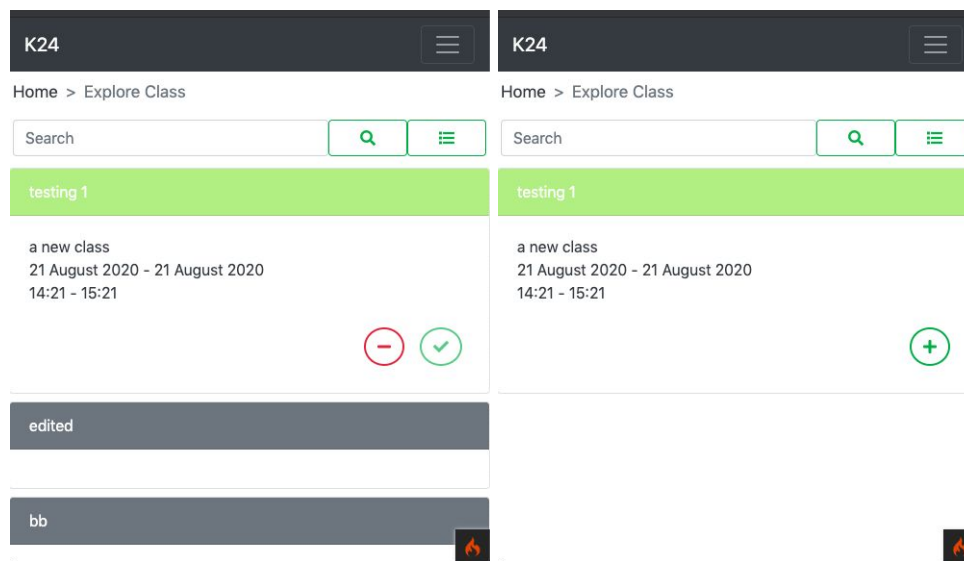
Bisa dilihat: Admin, Host, Peserta

Saat explore kelas, admin akan diperlihatkan kelas yang masih aktif dan yang sudah berakhir. Kelas yang masih aktif diberikan warna dan kelas yang telah berakhir diberi warna abu dan panel dikecilkan. Tetapi Host dan Peserta hanya diperlihatkan kelas yang masih aktif. Search bar juga bisa digunakan untuk mencari kelas tertentu. Untuk mendaftar ke kelas, user bisa menekan tombol “+” dan jika jadwalnya masih kosong saat itu, user bisa mendaftar dan diperlihatkan alert ‘sukses’. Tetapi, jika user sudah mempunyai kelas lain saat kelas itu, user akan diperlihatkan alert ‘Time conflict’. Jika user sudah terdaftar kelas, tombol “+” akan digantikan tombol inaktif “✓”. Selain itu, kelas yang tanggal mulai sudah lewat akan tetap muncul di list kelas tetapi tombol “+” akan dimatikan dan user lain sudah tidak bisa mendaftar lagi. Selain tombol “+” dan “✓”, Admin bisa menggunakan tombol “-” untuk menonaktifkan

kelas. Panel kelas bisa dipilih user untuk melihat detail kelas masing-masing, tetapi kelas aktif dan kelas non-aktif akan memperlihatkan info berbeda.

Backend: list kelas ditarik dari tabel 'classes' dan tanggal setiap kelas di cek agar di update statusnya di tabel 'classes'. Jika kelas didaftar "+", Controller/Classes akan cek jika user itu bisa mengikuti kelas tersebut (jika kelas aktif, sudah mulai, jadwal bebas. etc.). Jika bisa mendaftar, tabel 'user_classes' akan ditambah entry dan tabel 'user_scores' akan ditambah entry untuk pretest_id dan posttest_id dengan user_id dan class_id kelas tersebut. Jika kelas di remove "-" oleh Admin, kelas akan di nonaktifkan (status diganti 2 di tabel 'classes').

Files: Controller: Classes.php, Models: ClassModel.php. Views: header.php, footer.php, exploreClasses.php



View Class

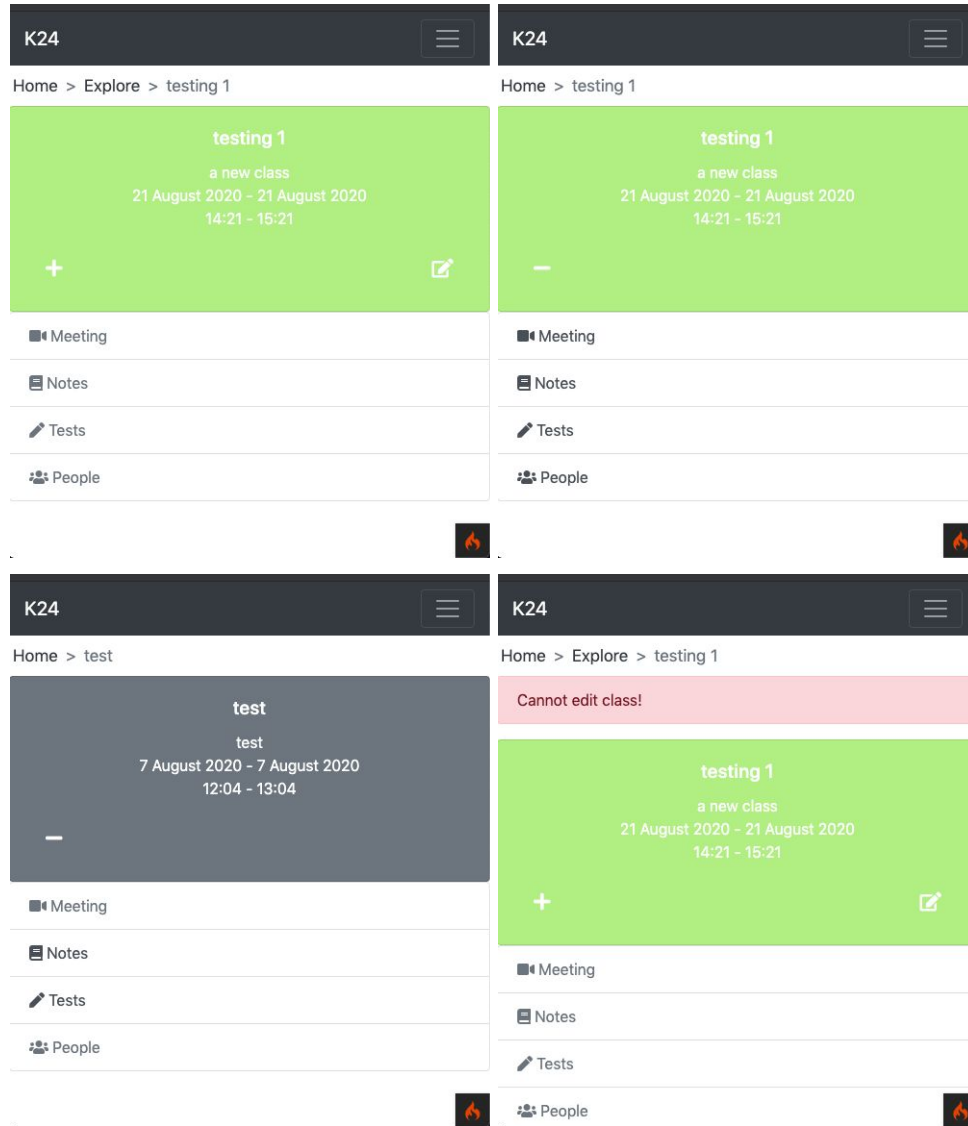
Bisa dilihat: Admin, Host, Peserta

Setelah kelas panel dipilih, user bisa melihat detail kelas tersebut. Jika user sudah terdaftar kelas itu, ia akan bisa mengakses Meeting, Notes, Tests dan People dan bisa keluar dari kelas dengan tombol "-" (image atas kanan). Kalau user belum terdaftar, dia hanya bisa melihat detail kelas (Image atas kiri) dan mendaftar dengan tombol "+", tetapi tombol Meeting, Notes, Tests dan People dimatikan. Jika user adalah host kelas ini atau Admin, dia bisa mengedit kelas dengan tombol 'edit' (image atas kiri). Tetapi, jika user bukan host kelas ini atau Admin, alert error akan muncul jika user mencoba edit kelas (image bawah kanan). Setelah kelas sudah berakhir, panel akan berwarna abu dan user hanya bisa mengakses Notes dan Tests (Nilai tes), user sudah tidak bisa join meeting dan melihat peserta lain (image bawah kiri).

Backend: Saat view class, Classes/Controller set Class Session dengan info kelas, pretest_id, posttest_id, isHost dan joined. *isHost* = 1 jika user adalah host kelas ini dan 0 jika user hanya peserta kelas. *joined* = 1 jika user sudah terdaftar kelas ini dan 0 jika belum terdaftar. Info ini bisa di cek melalui tabel 'user_classes'. User akan diperlihatkan tombol sesuai dengan status *isHost* dan *joined*. **Hanya host kelas bisa mengganti/edit kelas! Meskipun user mempunyai**

role Host, jika ia bukan host kelas ini, ia tidak bisa mengedit info kelas, membuat test, menambah note, etc. dan diberikan permisi role Peserta saat melihat atau mengakses info kelas ini.

Files: Controller: Classes.php, Models: ClassModel.php, ManagePermission.php. Views: header.php, footer.php, viewClass.php



Edit Class

Bisa dilihat: Admin, Host Kelas

Host kelas bisa mengganti info kelas. Misalnya, deskripsi kelas, waktu kelas atau link video call. Setelah info diganti, user perlu save. Alert sukses akan muncul jika info telah di update.

Backend: info kelas di update di tabel 'classes'

Files: Controller: Classes.php, Models: ClassModel.php, ManagePermission.php. Views: header.php, footer.php, editClass.php

Edit Class

Class Name

Description

Call link

Start Time

End Time

Recurring

☒ 1 day

☐ Weekly



Meeting

Bisa dilihat: Admin, Host, Peserta

Tombol meeting langsung membawa user ke link yang diberikan di info kelas.

Backend: list item di wrap <a> tag yang membawa user ke 'call_link' URL di tabel 'classes' untuk kelas tersebut.

Files: Controller: Classes.php, Models: ClassModel.php, ManagePermission.php. Views: header.php, footer.php, viewClass.php

Notes

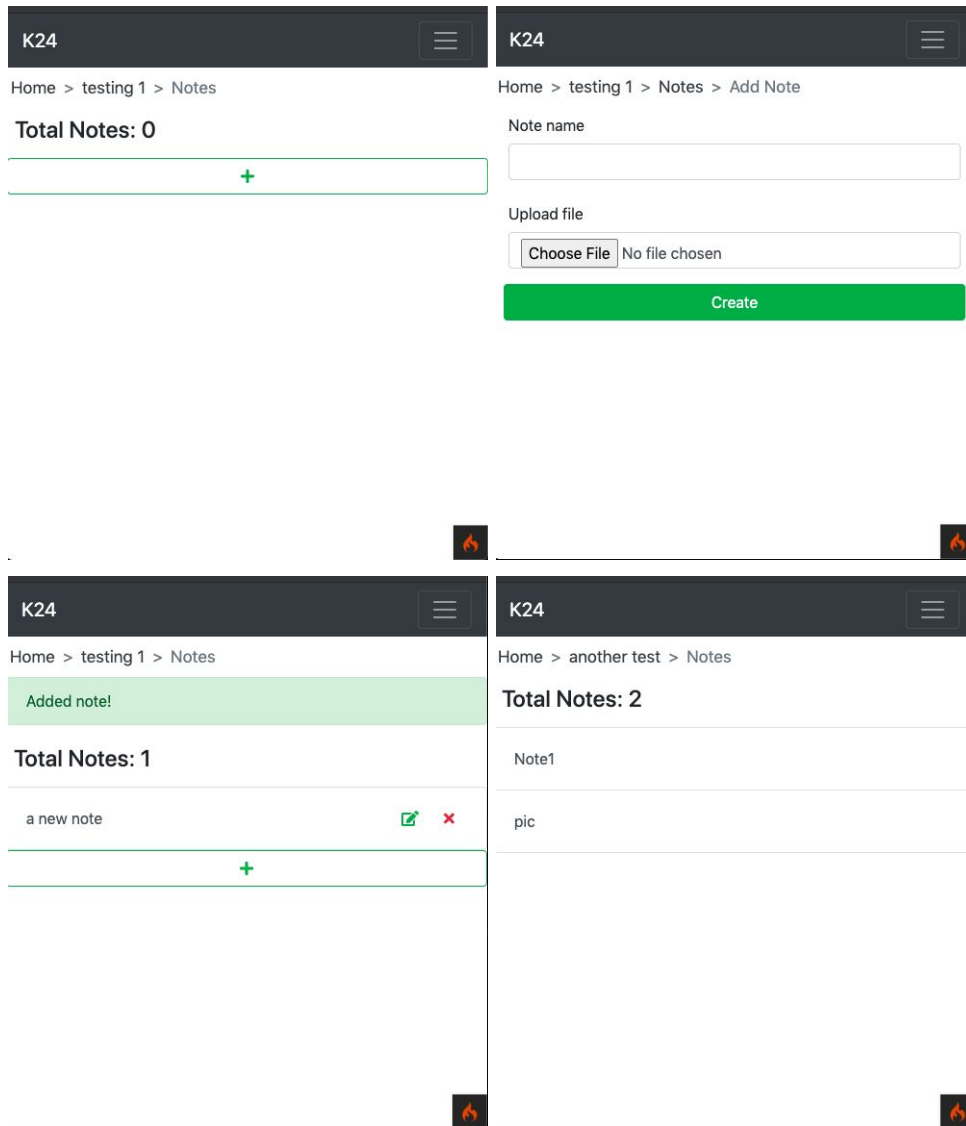
Bisa dilihat: Admin, Host, Peserta

Add, Edit & View Note

Host kelas bisa upload note baru dengan tombol "+". Untuk menambah note, host hanya perlu memberi note nama, mengupload file note dan save. Jika note telah dibuat, host bisa melihat aler sukses 'Added note!'. Setelah note di upload, host bisa edit note ini dengan mengupload file baru atau mengganti nama note. Setelah itu, peserta kelas bisa langsung melihat note melalui browser.

Backend: Adding new note membuat entry baru di tabel 'notes' dan edit note update info di tabel 'notes'. Kemudian, list item di wrap <a> tag ke uploads folder dan kolom 'path' di tabel 'notes' agar user langsung dibawa ke file saat list item dipilih.

Files: Controller: Notes.php, Models: NoteModel.php, ManagePermission.php. Views: header.php, footer.php, notes.php, addNote.php, editNote.php



People

Bisa dilihat: Admin, Host, Peserta

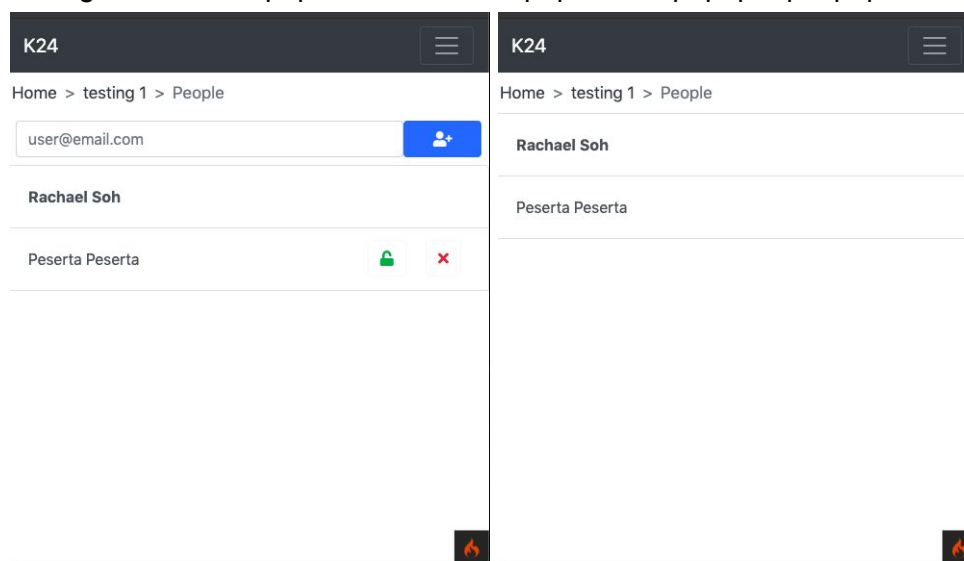
Invite User & View people

People page memperlihatkan user lain yang terdaftar dalam kelas ini. Jika user adalah host kelas namanya akan di bold dan user yang role Peserta namanya tidak bold. Page people Peserta akan seperti image kanan karena mereka hanya bisa melihat nama peserta lain. Tetapi, host kelas akan melihat page seperti image kiri karena mereka bisa invite user lain untuk mendaftar ke kelas ini. Host just bisa request peserta menjadi host dengan tombol lock hijau dan remove user dari kelas dengan tombol “x” merah. Setelah host request disetujui oleh Admin, peserta tersebut akan menjadi host kelas juga. Host harus invite user melalui email

mereka. Host akan mengisi input field dengan email user yang ingin di invite ke kelas ini dan memilih tombol biru “invite user”. Email invite akan dikirim ke email dengan link kelas tersebut dan jika email terkirim, akan muncul alert ‘sukses’ atau alert ‘fail’ jika email tidak terkirim.

Backend: Jika user di remove oleh host, entry ‘user_classes’ user dan kelas tersebut akan dihapus, tetapi user bisa diinvite lagi dan test score user tetap ada, status tinggal diaktifkan lagi. Kalau user di request menjadi host, host request akan ditambahkan ke tabel ‘hostRequest’ dan diberikan status 1 (pending), setelah disetujui atau ditolak admin, request ini akan di update jadi 0 dan kolom ‘host’ di tabel ‘user_classes’ akan diganti sesuai dengan action Admin. Untuk invite user baru, app/Config/Email.php perlu diganti untuk memberikan akses ke email account yang akan mengirim email invite ini, selain itu email message akan ada link ke kelas tersebut.

Files: Controller: Classes.php, Models: ClassModel.php, ManageUser.php, ManagePermission.php. Views: header.php, footer.php, people.php



Tests

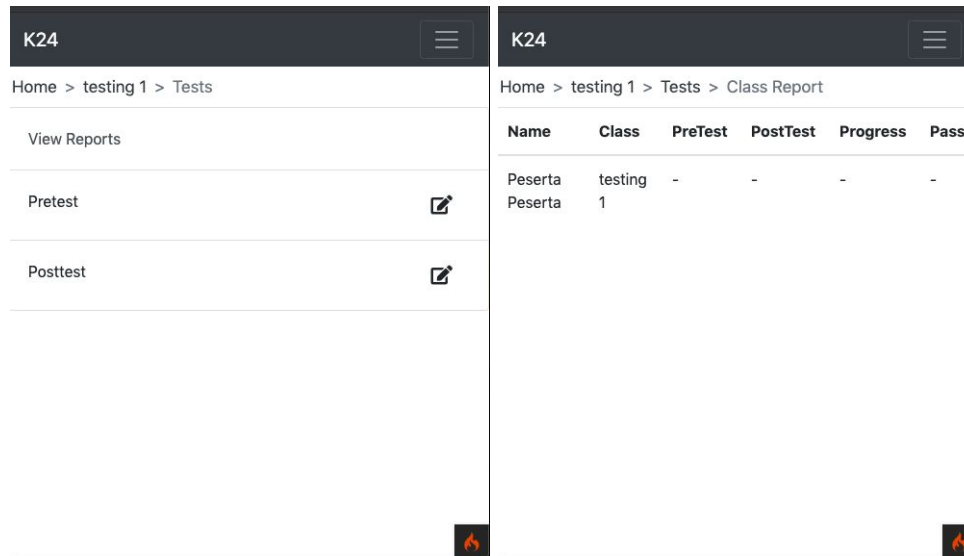
Bisa dilihat: Admin, Host, Peserta

Tests page

Test page host kelas atau Admin akan berbeda dengan Test page Peserta. Host atau Admin bisa melihat nilai tes semua peserta kelas dan juga bisa mengedit pretest dan posttest. Nilai Peserta yang terdaftar kelas itu akan diperlihatkan secara tabel. Rapor kelas akan ada nama peserta, kelas, nilai pretest, nilai posttest, jika angka naik (Progress) dan jika peserta lulus (Pass) .

Backend: class session akan di cek jika *isHost* = 1. Kalau user adalah host kelas, ia akan melihat ‘View Reports’ dan opsi edit tes. Rapor kelas langsung ditarik dari tabel ‘user_scores’ dan diperlihatkan sebagai tabel. (Lihatlah Model TestModel `getClassReports` untuk melihat SQL Query yang format nilai seperti yang diperlihatkan di page ‘Class Report’)

Files: Controller: TestReport.php, Models: TestModel.php, ClassModel.php, ManagePermission.php. Views: header.php, footer.php, tests.php



Edit Test & Edit Question

Bisa dilihat: Admin, Host Kelas

Host bisa mengedit test dengan page ini. Pertama, host perlu set durasi Pretest (kiri) dan tanggal, waktu mulai dan waktu akhir posttest (kanan). **Sebelum info ini diisi dan di save oleh host, Peserta tidak bisa melakukan test.** Setelah info ini ditentukan, host bisa menambah soal dengan tombol "+". Host akan dibawa ke page 'Edit Question' (kiri bawah) dan soalnya langsung di input the field 'Question'. Kemudian, pilihan jawaban bisa ditambah dengan tombol hijau 'Add more'. Input field opsi kemudian diisi dan jawaban soalnya dipilih dengan checkbox di kolom 'Answer'. Setiap soal bisa diberi jawaban sebanyak mungkin, setiap pilihan yang dicentang akan memberikan 1 poin ke Peserta. Untuk menghapus opsi, menggunakan tombol merah di kolom 'Remove' dan save. Setelah soalnya selesai di edit, tekanlah tombol biru 'Submit' dan akan keluar alert sukses 'Question edited!' setelah info di update. Kemudian, host bisa balik ke page 'Test Editing' (kanan bawah) dan menambah soal lain, mengedit soal yang sudah dibuat (dibawa balik ke page Edit Question), menghapus soal yang sudah dibuat atau mengganti durasi/tanggal test. Score total tes bisa dilihat di 'Total Questions: ' row. Misalnya, tes ini hanya ada 1 soal jadi nilai totalnya hanya 1. (kanan bawah)

Backend: kolom *duration*, *test_date*, *start_time*, *end_time* di tabel 'tests' akan di update setelah diisi host. Saat tambal soal, entry baru akan dibuat di tabel 'questions' dengan *question_id* yang baru dan soalnya disimpan di kolom *question*. Saat menambah pilihan jawaban dengan 'Add more', entry baru akan dibuat di tabel 'options' dengan *option_id* yang baru dan opsi disimpan di kolom *option_desc*. Jika checkbox dicentang, opsi itu akan diberikan value 1 di kolom *ans*. *Total_score* akan di update di tabel 'tests' untuk setiap soal yang ditambah ke tes, setiap soal akan dianggap sebagai 1 poin (5 soal = 5 poin total). *Total_score* akan diperlihatkan di row 'Total Questions' di view editTest.php. Info tes, soal dan opsi semua bisa di edit melalui pages 'Edit Test' dan 'Edit Question' dengan mengakses komponennya langsung dari tabel dengan *test_id*, *question_id* atau *option_id*.

Files: Controller: TestReport.php, Questions.php. Models: QuestionModel.php, TestModel.php, ClassModel.php, ManagePermission.php. Views: header.php, footer.php, editTest.php, editQuestion.php

K24

Home > testing 1 > Tests > Test Editing

Duration
Menit

Total Questions: 0

+

+

K24

Home > testing 1 > Tests > Test Editing

Test Date:

Starting Time:

Ending Time:

Total Questions: 0

+

+

K24

Home > testing 1 > Tests > Edit test > Edit Question

Question edited!

Question

#	Option	Answer	Remove
1	<input type="text" value="2"/>	<input checked="" type="checkbox"/>	<div>✖</div>
2	<input type="text" value="3"/>	<input type="checkbox"/>	<div>✖</div>
3	<input type="text" value="5"/>	<input type="checkbox"/>	<div>✖</div>

Add More

Submit

K24

Home > testing 1 > Tests > Test Editing

Question edited!

Duration
Menit

Total Questions: 1

1+1?

✎ ✕

+

+

Begin Test

Bisa dilihat: Peserta

Page test Peserta akan langsung memperlihatkan nilai tes pretest dan posttest diri sendiri dan opsi melakukan pretest dan posttest (Kiri atas). **Pretest bisa dilakukan tanpa batas kali diulang**, tetapi durasi tes tetap ditentukan oleh Host kelas. Sedangkan, **posttest hanya bisa dilakukan sekali pada waktu tertentu. Peserta tidak bisa mulai mengerjakan posttest sampai tanggal dan waktu yang ditentukan oleh host** (kanan atas).

Backend: dalam tabel 'tests', pretest diberi *test_status* 1 untuk berindikasi bahwa tes ini bisa dilakukan tanpa batas dan posttest diberi *test_status* 2 untuk berindikasi bahwa tes ini hanya bisa dilakukan sekali. Setelah pretest dilakukan, tes akan tetap berstatus 1. Tetapi, setelah

posttest dilakukan, *test_status* akan diganti jadi 3. *Test_status* bernilai 3 artinya tes sudah tidak bisa dilakukan lagi. Semua tes setelah kelas menjadi tidak aktif akan langsung diberi status 3 agar tidak bisa dilakukan lagi oleh Peserta.

Files: Controller: TestReport.php. Models: QuestionModel.php, TestModel.php. Views: header.php, footer.php, startTest.php

The image displays two side-by-side screenshots of a web application interface for a test system. Both screenshots have a dark header with 'K24' and a menu icon. The left screenshot shows the 'Tests' page with a breadcrumb 'Home > testing 1 > Tests'. It features two score boxes: 'PreTest Score: /100' and 'PostTest Score: /100'. Below these are two rows, 'Pretest' and 'Posttest', each with a pencil icon for editing. The right screenshot shows the 'Pretest' page with a breadcrumb 'Home > testing 1 > Tests > Pretest'. It has a pink error message 'Test not ready!'. Below this is a section titled 'Beginning Pretest:' containing fields for 'Date:', 'Start Time:', 'End Time:', 'Duration: 0 Minutes', and 'Total Score: 0'. At the bottom of this section is a blue button labeled 'Begin test'.

Take Test & Submit Exam

Bisa dilihat: Peserta

Setelah tes bisa dikerjakan (Image kanan atas), 'Duration' dan 'Total Score' akan kelihatan nomor selain 0 dan user bisa langsung mulai tes dengan tombol 'Begin test'. Untuk pretest, peserta bisa langsung mulai kapan saja. Tetapi, Posttest hanya bisa dimulai saat tanggal dan waktu mulai tertentu. Setelah tes dimulai, Pretest dan Posttest akan ada countdown durasi tes. Peserta hanya perlu memilih 1 pilihan per soal dan memilih tombol 'Submit Question'. Setelah sampai soal terakhir, Peserta bisa memilih tombol 'Submit Exam' untuk menyelesaikan tes.

Peserta tidak bisa balik ke soal sebelumnya setelah submit soal. Jika waktunya habis sebelum peserta selesai, Peserta akan langsung dibawa ke page 'Submitted' dan nilai tesnya akan dihitung sampai soal yang ia terakhir kerjakan. Saat di page 'Submitted', nilai tes akan langsung kelihatan sebagai persentase (nilai total 100). Kemudian, Peserta harus memilih tombol 'Back to class' agar dibawa balik ke Tests Page kelas tersebut dan nilai di Tests page langsung di update jadi kelihatan nilai tes yang baru dikerjakan, jika ia pretest atau posttest. **Kalau Peserta memulai Posttestnya telat, waktu tidak akan ditambah dan peserta hanya mempunyai waktu sampai waktu akhir yang ditentukan dari awal.** Jika Peserta tidak mengerjakan Posttest dalam jangka waktu yang tertentu, ia akan langsung diberi nilai 0 saat kelasnya berakhir (status = 2) di rapor kelas. Tetapi jika peserta tidak mengerjakan Pretest, nilai pretest akan tetap kosong.

Backend: setiap jawaban user untuk sebuah tes akan dimasukan tabel 'user_test_answers'. Kalau tes sudah pernah dilakukan (pretest), entry jawaban kolom *marks* hanya akan diupdate. Tetapi, kalau tes belum pernah dilakukan, entry jawaban baru akan dibuat. Setelah user 'Submit

Exam', nilai akan dihitung dengan menambahkan poin di kolom *marks* di tabel 'user_test_answer' dan persentase akan disimpan di kolom *score* tabel 'user_scores' untuk user itu. Entry nilai user untuk pretest dan posttest sebuah kelas langsung dibuat saat user mendaftar dengan kelas, jadi setelah user 'Submit exam', entry nilai untuk tes itu hanya perlu di update. Pretest diberi *test_status* = 1 dan Posttest diberi *test_status* = 2 di tabel 'user_scores'. Saat user bermulai pretest, *score* di reset jadi 0, jadi kalau user kehabisan waktu saat melakukan tes, soal yang tidak dapat dilakukan akan diberi poin 0 daripada poin dari tes yang dilakukan sebelumnya dan nilai terbaru yang disimpan. Kalau tes belum dilakukan, kolom *score* akan diberi value NULL dan kalau kelas sudah selesai dan masih belum dilakukan, pretest dibiarkan NULL dan posttest diberi nilai 0 dan dianggap tidak lulus.

Files: Controller: TestReport.php, Questions.php. Models: QuestionModel.php, TestModel.php.
Views: header.php, footer.php, takeTest.php, submitTest.php

The image displays three screenshots of a web application interface, likely for a test submission and results page. The interface is divided into three main sections, each with a dark header bar labeled 'K24' and a hamburger menu icon.

Top Left Screenshot: Shows the 'Time Left: 0h 0m 50s' and a question '1. 1+1'. Below the question are three radio button options: 2 (selected), 3, and 4. A green 'Submit Exam' button is visible at the bottom.

Top Right Screenshot: Shows the 'Submitted test!' message and the 'Test Score: 100 / 100'. A 'Back to class' button is located below the score.

Bottom Screenshot: Shows the 'Submitted test!' message and a comparison of scores. The 'PreTest Score' is '100/100' and the 'PostTest Score' is '/100'. Below the scores are two rows: 'Pretest' and 'Posttest', each with a pencil icon for editing.

Tools:

- Github Repository Online-Learning-webapp:
<https://github.com/rachael-soh/Online-Learning-webapp.git>
- CodeIgniter 4 Starter Files: <https://codeigniter.com/download>
- Login system: <https://www.youtube.com/watch?v=LC9GaXkdxF8>
- FullCalendar: <https://fullcalendar.io/docs/getting-started>. Download fullcalendar-5.3.0.zip
- XAMPP: <https://www.apachefriends.org/download.html>