

ДОКУМЕНТАЦИЈА

WireFrame

<http://wireframe.codeus.me/>

WEB PROGRAMIRANJE PHP 1

Лазар Лаловић 62/20

Садржај

УВОД	5
1.1 Коришћени програмски језици	5
1.2 Опис функционалности	5
1.3 Дизајн сајта	6
Странице на админ панелу	12
1.4 Организација сајта и база података	15
sitemap.xml	16
Кодови и синтакса	20
admin/controllers	20
add_survey.php	20
index.php	22
messages.php	23
modify.php	25
posts.php	27
surveys.php	28
users.php	29
admin/views	30
index.view.php	30
head.php	31
footer.php	31
sidenav.php	32
admin/views/content	33
index.view.php	33
messages.view.php	35
modify.view.php	38
posts.view.php	40
surveys.view.php	42
users.view.php	44
view-messages.view.php	46
config	48

config.php	48
main.....	48
index.php.....	48
routes	49
routes.php.....	49
core.....	51
bootstrap.php.....	51
Request.php.....	52
Router.php.....	53
Utilities.php.....	55
core/database	57
Connection.php	57
QueryBuilder.php.....	58
views/authentication	78
login.view.php.....	78
registration.view.php.....	80
verify.view.php	84
views/pages	85
author.view.php.....	85
contact.view.php	87
index.view.php.....	90
posts.view.php.....	93
views/posts	98
create-post.view.php	98
edit-post.view.php.....	101
view-post.view.php	105
views/users	107
profile.view.php	107
survey.view.php	112
views/partials	115
head.php	115
footer.php	116
navigation.php.....	121
post-comment.php.....	125

controllers	132
author.php	132
index.php	132
surveys.php	132
contact.php	133
controllers/validations	135
verify.php	135
controllers/user	136
profile.php	136
survey.php	137
update-image.php	138
controllers/authentication	139
login.php	139
logout.php	142
registration.php	142
controllers/posts	148
comment.php	148
create-post.php	149
delete-comment.php	152
delete-post.php	153
edit-post.php	155
posts.php	159
search.php	161
view-post.php	162
CSS	163
style.css	163
JavaScript	182
main.js	182

УВОД

1.1 Коришћени програмски језици

За израду сајта, коришћено је следеће:

- HTML
- CSS/SCSS (Materialize framework)
- JavaScript
- PHP
- PHP Mailer
- CKEditor
- Composer
- XML
- JSON
- Регуларни изрази

Сви наведени језици, библиотеке и фрејмворкови су писани унутар „Visual Studio Code” окружења.

1.2 Опис функционалности

Садржај свих страница се исписује динамички, изузев почетне „index” странице која је статичка (осим поруке добродошлице) и странице о аутору. Сајт је прављен дјелимично по узору на „MVC” структуру и уз сопствени „mini” фрејмворк. Главне функционалности које се требају навести јесу:

- Динамичко исписивање навигационог менија и видљивост по улогама
- Логовање и регистрација корисника, сви параметри се провјеравају и на клијентској и на серверској страни уз регуларне изразе
- Логовање и корисничким именом, а и мејлом
- Валидација регистрованог корисника путем „e-mail-a” верификационим токеном
- Контактирање администратора (у случају да је ауторизован корисник, аутоматски се попуњавају одређена поља)
- Могућност промјене профилне фотографије
- Анкетирање и памћење резултата (видљиво само администраторима)
- Креирање објава, које обавезно имају насловну фотографију, једну категорију(минимум) или више и само тијело објаве, које се може форматирати уз „CKEditor”
- Коментарисање и одговарање на коментаре
- Рутер, који делегира захтјеване „web” локације, те спријечава унос локација које програмер није дефинисао
- Претрага објава AJAX-ом и аутоматско ажурирање резултата, филтрирање по категоријама

- Управљање садржајем сајта из админ-панела. Корисници са администраторским улогама могу:
 - Провјерити цјелокупно стање на сајту, број корисника, објава и сл.
 - Читати поруке (идентификација прочитане/непрочитане поруке)
 - Додавати/брисати категорије (брисати само оне које нису повезане са ниједном објавом)
 - Креирати анкете
 - Деактивирати текуће, постојеће анкете
 - Одлучити која ће објава бити одобрена и приказана

1.3 Дизајн сајта

За израду сајта није коришћено готово рјешење, све је креирао аутор сајта. Прилагођен је за све врсте уређаја.

слика 1 . index



Welcome to WireFrame

The Developer's Blog to Keep You at the Top of Your Game.



Stay informed!

The field of software engineering is constantly developing. (No dum tse) With tech, frameworks, and programming best practices changing every day - even the best developers risk falling behind.

Nobody really wants to dedicate large chunks of their time to continued education. We get that. Fortunately, the internet is littered with developer blogs. And, while we here at WireFrame are anti-filtering, we are definitely pro-innovation, pro-growth and pro-learning from the trials and errors of others so we don't have to waste time ourselves.

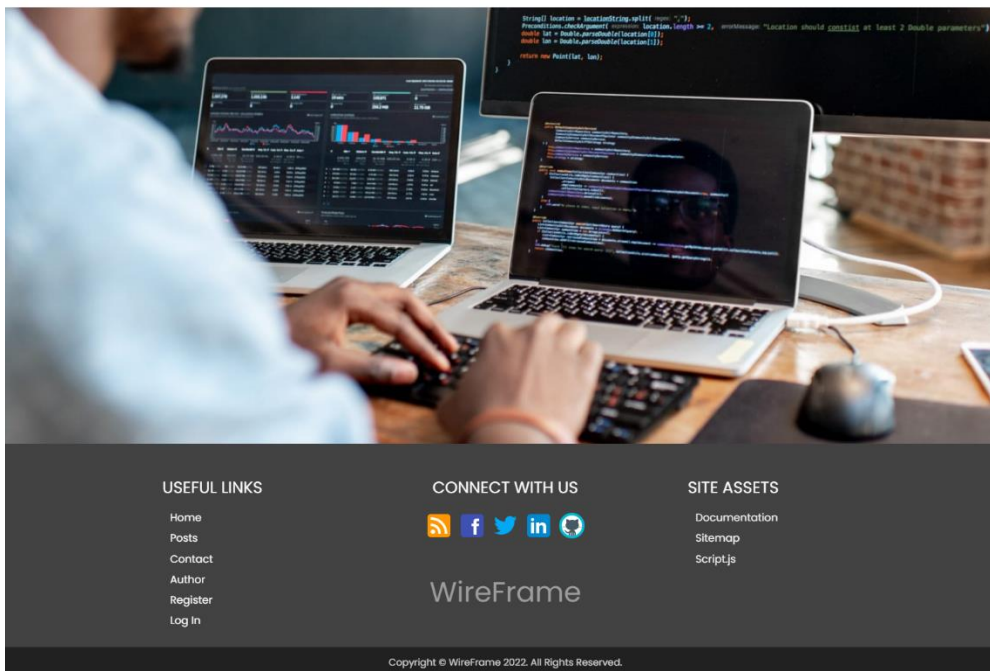


Interested in writing?

Simply create a profile and start writing programming topics, to help future generations!

Join us, and we will change and create the web together!

[REGISTER](#)



USEFUL LINKS

[Home](#)
[Posts](#)
[Contact](#)
[Author](#)
[Register](#)
[Log In](#)

CONNECT WITH US

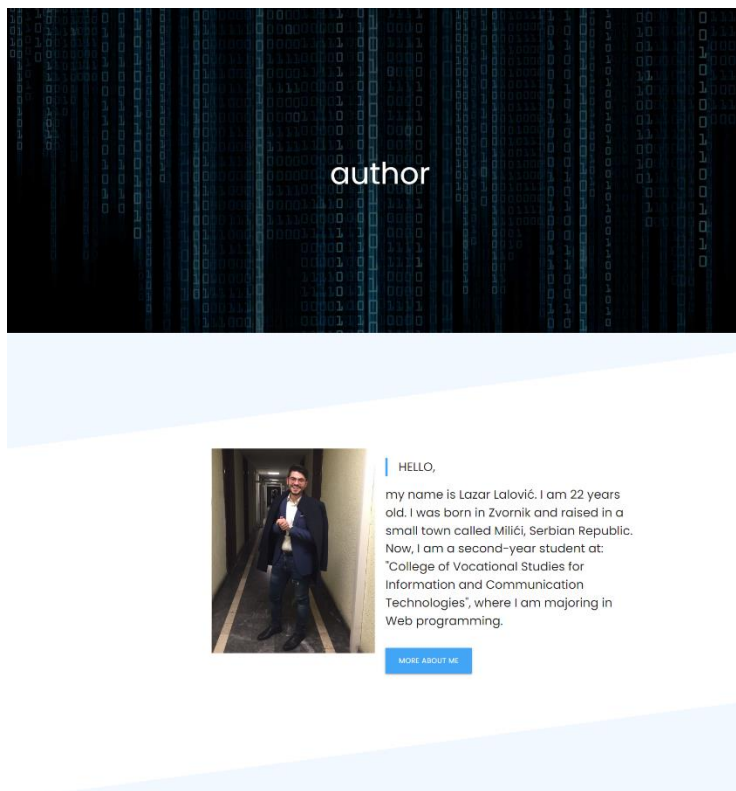


WireFrame

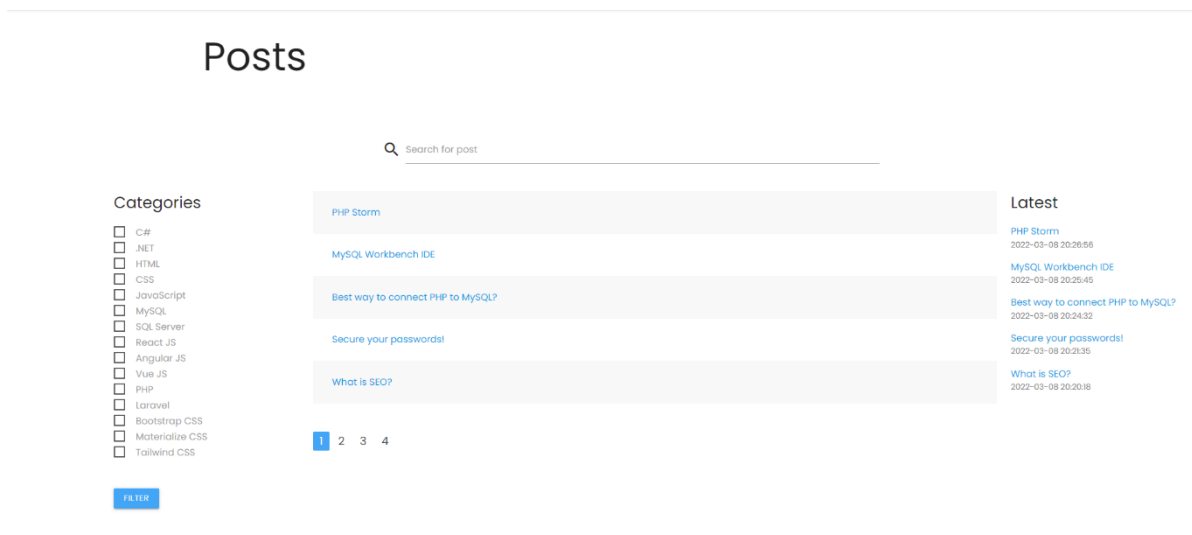
SITE ASSETS

[Documentation](#)
[Sitemap](#)
[Scripts](#)

Copyright © WireFrame 2022. All Rights Reserved.



слика 2. author



слика 3. posts

Hello, write us a message!

E-Mail	Message Title
Write us something...	

SEND

слика 4. contact

Register

First name
e.g. John

Last name
e.g. Doe

Username
e.g. johndoe

E-Mail
e.g. johndoe@example.com

Password
Min. 8 chars, one capital, one symbol (max. = 20)

Re-enter password
Repeat password

REGISTER

Already have an account? [Log In](#)

слика 5. register

Log In

Username or E-mail
Type here ...

Password
Enter your password ...

LOG IN

Don't have an account? [Register](#)

слика 6. login



Lazar Lalovic

Username: laxon00
 Mail: lazarlalovic@gmail.com
 Joined: 2022-03-08 11:21:07
 Role: Administrator

Lazar's posts

Postid	Post title
1	What's new in Laravel 9.0?
2	Laravel Routing
3	What is SEO?
4	Secure your passwords!
5	Best way to connect PHP to MySQL?

слика 7. profile (visitor)

Изглед профилне странице из перспективе власника профила (слика 8.) и посјетиоца на профил (слика 7.)

Јасно можемо примјетити поруке које се приказују посјетиоцима и власнику профила и функционалности.

Hello, Lazar!



Username: laxon00
 Mail: lazarlalovic@gmail.com
 Joined: 2022-03-08 11:21:07
 Role: Administrator
 Change profile picture:

+ My posts

Postid	Post title	Edit	Remove
1	What's new in Laravel 9.0?		
2	Laravel Routing		
3	What is SEO?		

слика 8. profile (owner)

What's new in C# 10?

Written by: [Luka Lukic](#) on: 2022-03-08 20:17:44

DELETE POST



What's new in C# 10

C# 10 adds the following features and enhancements to the C# language:

- Record structs
- Improvements of structure types
- Interpolated string handlers
- Global using directives
- File-scoped namespace declaration
- Extended property patterns
- Improvements on lambda expressions
- Allow const interpolated strings
- Record types can seal ToString()
- Improved definite assignment
- Allow both assignment and declaration in the same deconstruction
- Allow AsyncMethodBuilder attribute on methods
- CallerArgumentExpression attribute
- Enhanced #line pragma


C# 10 is supported on .NET 6. For more information, see [C# language versioning](#).
 You can download the latest .NET 6 SDK from the [.NET downloads page](#). You can also download [Visual Studio 2022](#), which includes the .NET 6 SDK.

Record structs

You can declare value type records using the [record struct](#) or [readonly record struct](#) declarations. You can now clarify that a record is a reference type with the record class declaration.

слика 9. post-view


Comments



Lazar Lalovic
 2022-03-09 00:51:05

You should consider switching to Laravel and PHP, this is useless...


Reply Delete



Luka Lukic
 2022-03-09 01:04:41

You know nothing John Snow...

Delete



Nikola Ljubic
 2022-03-09 01:05:37

I agree with you!

Delete

What do you think?

COMMENT

слика 10. post-view komentari

Create post

Title
CSS Grid system...

Category
Categories:

Post body

IMAGE

Upload cover image (important)

CREATE POST

Edit post

Title
MySQL Workbench IDE

Category
Categories: MySQL

Post body

IMAGE 1_0ktUIXORWEjXLh7aNg4ELA.png

Upload cover image (important)

SAVE POST

Странице за креирање и измјену објаве.

Странице на админ панелу: dashboard/posts - Одобравање објава.

Dashboard > Posts				
PostId	PostTitle	PostAutor	CreatedAt	Approve
1	What's new in Laravel 9.0?	laxon00	2022-03-08 15:38:57	✓

dashboard/modify - Модификација категорија.

Dashboard > Modify

Add or remove categories

+

Add category
.NET|

+

CategoryId	Category name	Remove
1	C#	✗

dashboard/messages - Све поруке. (аутор је грешком отворио погрешну поруку... углавном зелена непочитана, сива прочитана...)

Dashboard > Messages

MessageId	Subject	From	Received At	Delete
8	Mozda otvorena	luka.lukic@ict.edu.rs	2022-03-06 23:41:15	✕
7	Nije otvorena poruka	milkacanic@dobrovecce.net	2022-03-06 23:40:55	✕
6	Procitana poruka	lazarialovic@gmail.com	2022-03-06 23:40:19	✕

EMPTY INBOX

dashboard/messages?view_message=id - Тијело поруке.

Dashboard > Messages > Nije Otvorena Poruka

BACK







DELETE

Title: Nije otvorena poruka

Ovo nije otvorena poruka!

Sent At: 2022-03-06 23:40:55, by milkacanic@dobrovecce.net

Dashboard > Users

	Username	Joined	Role
	laxon00	2022-03-08 11:21:07	Administrator
	cokolino1	2022-03-08 12:03:30	Authorized
	luka95	2022-03-08 12:04:09	Administrator
	linna95	2022-03-08 12:05:40	Authorized
	nikola87	2022-03-08 12:07:11	Administrator
	boban72	2022-03-09 00:18:04	Authorized

dashboard/users

-

Увид у кориснике и модификација улога.

dashboard/surveys - Анкетирање (лијево администраторски дио, десно кориснички)

Dashboard > Surveys

Add or deactivate surveys

+


SurveyId	Question	IsActive
1	How often do you visit us? - Sometimes - 1	✗
2	How often do you write posts? -	✓

Take our surveys

PollId	Question	Your answer
2	How often do you write posts?	<div><div><input type="radio"/> Never</div><div><input type="radio"/> Sometimes</div><div><input type="radio"/> Always</div></div> <div>SEND</div>


dashboard/index - Статистика

Dashboard > Index




6

USERS




20

TOPICS




4

COMMENTS




15

CATEGORIES



+300

MATERIALS

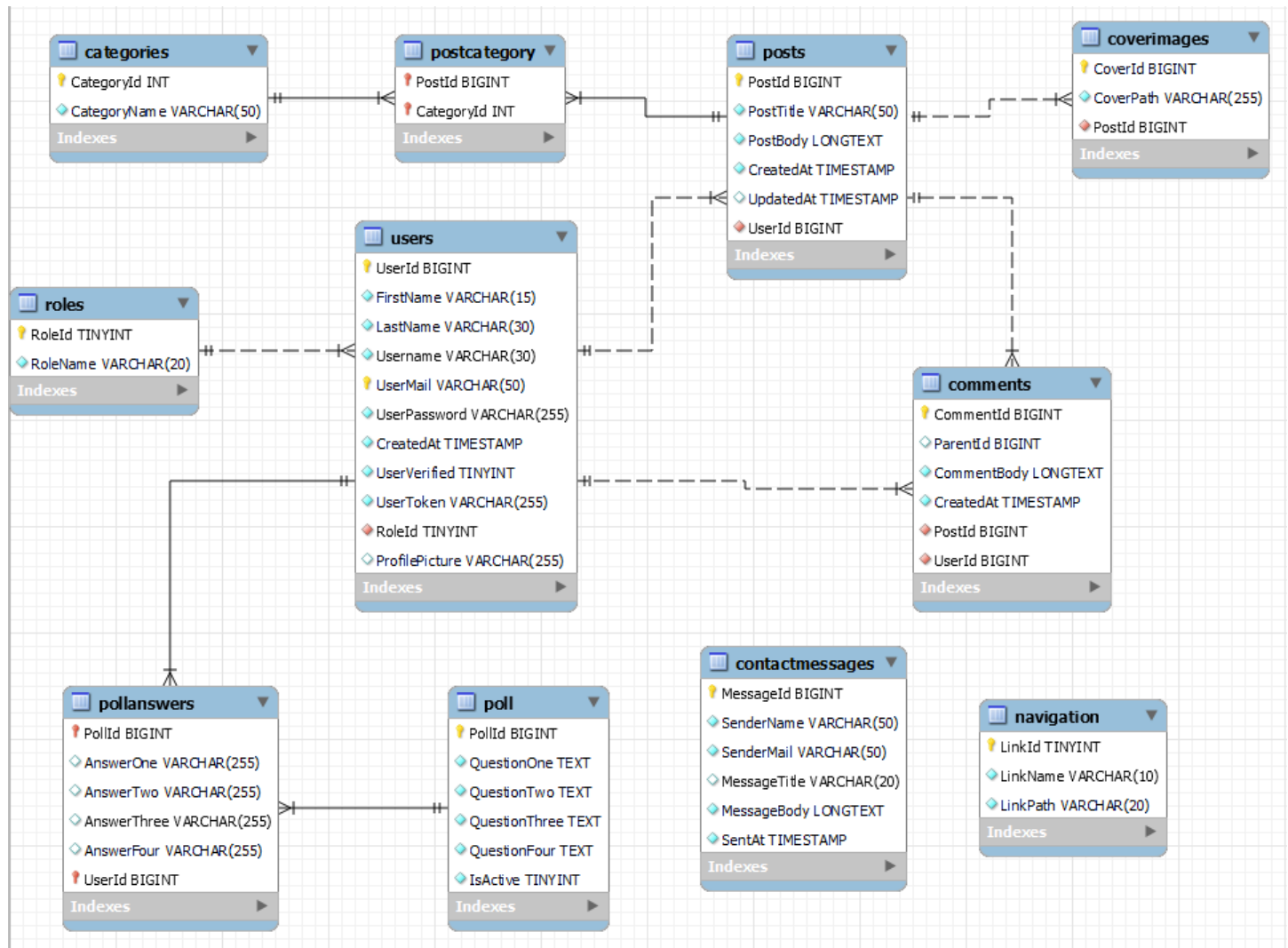


+10.000

VISITORS

1.4 Организација сајта и база података

Дијаграм базе података



Мапа сајта

sitemap.xml

```
<?xml version="1.0" encoding="utf-8" ?>

<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">

  <url>

    <loc>http://wireframe.codeus.me/posts</loc>

    <lastmod>2022-03-11</lastmod>

    <changefreq>daily</changefreq>

    <priority>1</priority>

  </url>

  <url>

    <loc>http://wireframe.codeus.me/profile</loc>

    <lastmod>2022-03-11</lastmod>

    <changefreq>daily</changefreq>

    <priority>0.9</priority>

  </url>

  <url>

    <loc>http://wireframe.codeus.me/dashboard/messages</loc>

    <lastmod>2022-03-11</lastmod>

    <changefreq>daily</changefreq>

    <priority>0.9</priority>

  </url>
```



```
<url>

  <loc>http://wireframe.codeus.me/dashboard/index</loc>

  <lastmod>2022-03-07</lastmod>

  <changefreq>weekly</changefreq>

  <priority>0.85</priority>

</url>

<url>

  <loc>http://wireframe.codeus.me/dashboard/users</loc>

  <lastmod>2022-03-09</lastmod>

  <changefreq>weekly</changefreq>

  <priority>0.8</priority>

</url>

<url>

  <loc>http://wireframe.codeus.me/index</loc>

  <lastmod>2022-03-09</lastmod>

  <changefreq>monthly</changefreq>

  <priority>0.75</priority>

</url>

<url>

  <loc>http://wireframe.codeus.me/dashboard/modify</loc>

  <lastmod>2022-03-01</lastmod>

  <changefreq>monthly</changefreq>
```

```
<priority>0.7</priority>

</url>

<url>

<loc>http://wireframe.codeus.me/dashboard/users</loc>

<lastmod>2022-03-02</lastmod>

<changefreq>monthly</changefreq>

<priority>0.7</priority>

</url>

<url>

<loc>http://wireframe.codeus.me/dashboard/surveys</loc>

<lastmod>2022-03-02</lastmod>

<changefreq>monthly</changefreq>

<priority>0.7</priority>

</url>

<url>

<loc>http://wireframe.codeus.me/contact</loc>

<lastmod>2022-03-01</lastmod>

<changefreq>yearly</changefreq>

<priority>0.65</priority>

</url>

<url>

<loc>http://wireframe.codeus.me/author</loc>
```

```
<lastmod>2022-03-03</lastmod>
```

```
<changefreq>yearly</changefreq>
```

```
<priority>0.5</priority>
```

```
</url>
```

```
</urlset>
```

Кодови и синтакса

admin/controllers

add_survey.php

```
<?php

$access_method = Request::getMethod();

session_start();

if($access_method !== 'POST' || !isset($_SESSION['user_id']) || $_SESSION['role'] !== 1)

Utilities::redirect('/index');

else

{

    $question = $_POST['add_question'];

    $users = $database->selectRecords('Users');

    $database->insertRecords('Poll', [

        'Question' => $question

    ]);

    $last_poll = $database->selectLastRecords('Poll', 'PollId', 1, 0);
```

```
$last_poll_id = $last_poll[0]->PollId;

foreach($users as $user)

{

    $database->insertRecords('PollAnswers', [

        'PollId' => $last_poll_id,

        'Answer' => null,

        'UserId' => $user->UserId

    ]);

}

Utilities::redirect('/dashboard/surveys');

}
```

index.php

```
<?php

$page_title = 'Dashboard';

$access_method = Request::getMethod();

session_start();

if($access_method !== 'GET' || empty($_SESSION['user_id']) || $_SESSION['role'] !== 1) Utilities::redirect('/index');

else

{

    $posts_count = $database->getRecordCount('Posts');

    $comment_count = $database->getRecordCount('Comments');

    $categories_count = $database->getRecordCount('Categories');

    $users_count = $database->getRecordCount('Users');

    require 'admin/views/pages/index.view.php';

}
```

messages.php

```
<?php

$page_title = 'Dashboard';

$access_method = Request::getMethod();

session_start();

if($access_method !== 'GET' || empty($_SESSION['user_id']) || $_SESSION['role'] !== 1) Utilities::redirect('/index');

else

{

    if(isset($_GET['delete_message']) && isset($_SESSION['user_id']) && $_SESSION['role'] === 1)

    {

        $message_id = $_GET['delete_message'];

        $database->deleteRecords('ContactMessages', 'MessageId', $message_id);

        Utilities::redirect('/dashboard/messages');

    }

    else if(isset($_GET['view_message']) && isset($_SESSION['user_id']) && $_SESSION['role'] === 1)

    {

        $message_id = $_GET['view_message'];

        $message = $database->selectFilteredRecord('ContactMessages', 'MessageId', $message_id);
```

```
        $database->updateRecord('ContactMessages', 'HasRead', 1, 'MessageId', $message_id);

        require 'admin/views/partials/content/view-message.view.php';

    }

    else if(isset($_GET['delete_all']) && isset($_SESSION['user_id']) && $_SESSION['role'] === 1)

    {

        $database->deleteAllRecords('ContactMessages');

        Utilities::redirect('/dashboard/messages');

    }

    else

    {

        $messages = $database->selectRecords('ContactMessages', 1);

        require 'admin/views/pages/index.view.php';

    }

}
```


modify.php

```
<?php

$access_method = Request::getMethod();

session_start();

if($access_method === 'GET' && isset($_SESSION['user_id']) && $_SESSION['role'] === 1)

{

    if(isset($_GET['delete_category']) && isset($_SESSION['user_id']) && $_SESSION['role'] === 1)

    {

        $category_id = $_GET['delete_category'];

        $post_count = $database->selectPostsByCategory([$category_id]);

        if(count($post_count)) Utilities::redirect('/dashboard/modify?exception=You must first delete all posts bound
to this category!');

        else

        {

            $database->deleteRecords('Categories', 'CategoryId', $category_id);

            Utilities::redirect('/dashboard/modify');

        }

    }

}
```

```
$categories = $database->selectRecords('Categories');

require 'admin/views/pages/index.view.php';

}

else

{

    $new_category = $_POST['add_category'];

    if(empty($new_category) || !preg_match('/^[a-zA-Z0-9#\s\.\.]{2,50}$/', $new_category))

Utilities::redirect('/dashboard/modify?error=Provide the name for new category!');

    else

    {

        $database->insertRecords('Categories', [

            'CategoryName' => $new_category

        ]);

        Utilities::redirect('/dashboard/modify');

    }

}
```

posts.php

```
<?php

$page_title = 'Dashboard';

$access_method = Request::getMethod();

session_start();

if(empty($_SESSION['user_id']) || $_SESSION['role'] !== 1) Utilities::redirect('/index');

else if($access_method === 'POST')

{

    $str_json = file_get_contents('php://input');

    $decoded_json = json_decode($str_json, false);

    $post_id = $decoded_json->post_id;

    $database->updateRecord('Posts', 'Approved', 1, 'PostId', $post_id);

}

else

{

    $posts = $database->selectFilteredRecords('Posts', 'Approved', 0);

    require 'admin/views/pages/index.view.php';

}
```

surveys.php

```
<?php

$page_title = 'Dashboard';

$access_method = Request::getMethod();

session_start();

if(empty($_SESSION['user_id']) || $_SESSION['role'] !== 1) Utilities::redirect('/index');

else if($access_method === 'POST')

{

    $str_json = file_get_contents('php://input');

    $decoded_json = json_decode($str_json, false);

    $survey_id = $decoded_json->survey_id;

    $database->updateRecord('Poll', 'IsActive', 0, 'PollId', $survey_id);

}

else

{

    $surveys = $database->selectRecords('Poll');

    require 'admin/views/pages/index.view.php';

}
```

users.php

```
<?php

$page_title = 'Dashboard';

$access_method = Request::getMethod();

session_start();

if(empty($_SESSION['user_id']) || $_SESSION['role'] !== 1) Utilities::redirect('/index');

else if($access_method === 'POST')

{

    $str_json = file_get_contents('php://input');

    $decoded_json = json_decode($str_json, false);

    $role_id = $decoded_json->role_id;

    $user_id = $decoded_json->user_id;

    $database->updateRecord('Users', 'RoleId', $role_id, 'UserId', $user_id);

}

else

{

    $users = $database->selectRecords('Users');

    $roles = $database->selectRecords('Roles');

    require 'admin/views/pages/index.view.php';

}
```

admin/views

index.view.php

```
<?php

$uri = substr(Request::getUri(), strpos(Request::getUri(), '/') + 1, strlen(Request::getUri()));

?>

<?php require 'admin/views/partials/main/head.php' ?>

<?php require 'admin/views/partials/main/sidenav.php' ?>

<div class="container">

    <nav class="mb-3">

        <div class="nav-wrapper blue lighten-1 pl-1">

            <div class="col s12">

                <a href="/dashboard/index" class="breadcrumb">Dashboard</a>

                <a href="/dashboard/<?= $uri ?>" class="breadcrumb page-name">

                    <?= $uri ?>

                </a>

            </div>

        </div>

    </nav>

    <?php require 'admin/views/partials/content/' . $uri . '.view.php' ?>

</div>

<?php require 'admin/views/partials/main/footer.php' ?>
```

head.php

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="/public/css/style.css" />

    <!-- Compiled and minified CSS -->

    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css" />

    <script type="text/javascript" src="/public/js/main.js" defer></script>

    <title>Dashboard</title>

</head>

<body>
```

footer.php

```
<!-- jQuery -->

<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-
/xUj+30JU5yEx1q6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>

<!-- Compiled and minified JavaScript -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>
```

```
<script>

    $(document).ready(function(){

        $('.sidenav').sidenav()

    })

</script>

</body>

</html>
```

sidenav.php

```
<?php

    $links = $database->selectRecords('Navigation');

?>

<ul id="slide-out" class="sidenav show-on-small-only grey lighten-3">

    <?php for($i = 0; $i < count($links); $i++) : ?>

        <li>

            <a href="<?= $links[$i]->LinkPath ?>">

                <?= $links[$i]->LinkName ?>

            </a>

        </li>

    <?php endfor ?>

    <li>

        <a href="/index">Return to WireFrame</a>
```



```
</li>

</ul>

<div class="row blue lighten-1 p-1 center">

  <a data-target="slide-out" class="sidenav-trigger left">

    <i class="material-icons white-text small">menu</i>

  </a>

  <b class="white-text">ADMIN PANEL v1.0.0</b>

</div>
```

admin/views/content

index.view.php

```
<div class="row info-cards">

  <div class="col s12 m5 l2 offset-l1 stats-card">

    <i class="material-icons blue-text text-lighten-1 medium">account_box</i>

    <p>

      <?= $users_count->Rows ?>

    </p>

    <p>Users</p>

  </div>

  <div class="col s12 m5 offset-m2 l2 offset-l2 stats-card">

    <i class="material-icons blue-text text-lighten-1 medium">speaker_notes</i>

    <p>
```

```
<?= $posts_count->Rows ?>

</p>

<p>Topics</p>

</div>

<div class="col s12 m5 l2 offset-l2 stats-card">

    <i class="material-icons blue-text text-lighten-1 medium">comment</i>

    <p>

        <?= $comment_count->Rows ?>

    </p>

    <p>Comments</p>

</div>

<div class="col s12 m5 offset-m2 l2 offset-l1 stats-card">

    <i class="material-icons blue-text text-lighten-1 medium">assessment</i>

    <p>

        <?= $categories_count->Rows ?>

    </p>

    <p>Categories</p>

</div>

<div class="col s12 m5 l2 offset-l2 stats-card">

    <i class="material-icons blue-text text-lighten-1 medium">get_app</i>

    <p>

        +300
```

```
</p>

<p>Materials</p>

</div>

<div class="col s12 m5 offset-m2 l2 offset-l2 stats-card">

  <i class="material-icons blue-text text-lighten-1 medium">mood</i>

  <p>

    +10.000

  </p>

  <p>Visitors</p>

</div>

</div>
```

[messages.view.php](#)

```
<table class="highlight centered mb-3">

  <thead>

    <tr>

      <td class="center">MessageId</td>

      <td class="center">Subject</td>

      <td class="center">From</td>

      <td class="center">Received At</td>

      <td class="center">Delete</td>
```

```
</tr>

</thead>

<tbody>

<?php if(!empty($messages)) : ?>

    <?php foreach($messages as $message) : ?>

        <tr class="<?= $message->HasRead ? 'grey lighten-2' : 'green lighten-5' ?>">

            <td>

                <?= $message->MessageId ?>

            </td>

            <td>

                <a href="/dashboard/messages?view_message=<?= $message->MessageId ?>">

                    <?= $message->MessageTitle ?>

                </a>

            </td>

            <td>

                <?= $message->SenderMail ?>

            </td>

            <td>

                <?= $message->CreatedAt ?>

            </td>

            <td>
```

```
<a href="/dashboard/messages?delete_message=<?= $message->MessageId ?>">

    <i class="material-icons red-text">close</i>

</a>

</td>

</tr>

<?php endforeach ?>

<?php else : ?>

    <tr>

        <td colspan="5">No messages found!</td>

    </tr>

<?php endif ?>

</tbody>

</table>

<?php if(!empty($messages)) : ?>

    <a href="/dashboard/messages?delete_all=1" class="btn red">

        EMPTY INBOX

    </a>

<?php endif ?>
```

modify.view.php

```
<?php if(isset($_GET['exception'])) : ?>

    <span class="p-3 right red-text red lighten-3 text-darken-3">

        <?= $_GET['exception'] ?>

    </span>

<?php endif ?>

<h4 class="mb-3">Add or remove categories</h4>

<div class="row">

    <button id="addCategoryBtn" class="btn blue lighten-1">

        <i class="material-icons">add</i>

    </button>

    <form action="/dashboard/modify" method="POST">

        <div id="addCategoryField" class="input-field col l6 right <?= empty($_GET['error']) ? 'hide' : '' ?>">

            <input id="addCategory" name="add_category" type="text" />

            <label for="addCategory">Add category</label>

            <button type="submit" class="btn blue lighten-1"><i class="material-icons">add</i></button>

            <?php if(isset($_GET['error'])) : ?>

                <span class="red-text">

                    <?= $_GET['error'] ?>

                </span>

            <?php endif ?>

        </div>

    </form>

</div>
```

```
</div>

</form>

</div>

<table class="highlight striped centered mb-3">

  <thead>

    <tr">

      <td class="center">CategoryId</td>

      <td class="center">Category name</td>

      <td class="center">Remove</td>

    </tr>

  </thead>

  <tbody>

    <?php foreach($categories as $category) : ?>

      <tr>

        <td>

          <?= $category->CategoryId ?>

        </td>

        <td>

          <?= $category->CategoryName ?>

        </td>

        <td>
```

```
<a class="delete-category" href="/dashboard/modify?delete_category=<?= $category->CategoryId ?>">

    <i class="material-icons red-text">close</i>

</a>

</td>

</tr>

<?php endforeach ?>

</tbody>

</table>
```

posts.view.php

```
<table class="highlight striped centered">

    <thead>

        <tr>

            <td class="center">PostId</td>

            <td class="center">PostTitle</td>

            <td class="center">PostAutor</td>

            <td class="center">CreatedAt</td>

            <td class="center">Approve</td>

        </tr>

    </thead>

    <tbody>

        <?php if(empty($posts)) : ?>
```



```
<tr>

    <td colspan="5">No pending requests found!</td>

</tr>

<?php else : ?>

    <?php foreach($posts as $post) : ?>

        <?php

            $writer = $database->selectFilteredRecord('Users', 'UserId', $post->UserId);

        ?>

        <tr>

            <td><?= $post->PostId ?></td>

            <td><?= $post->PostTitle ?></td>

            <td><?= $writer->Username ?></td>

            <td><?= $post->CreatedAt ?></td>

            <td>

                <i class="white approve-post material-icons green-text">check</i>

            </td>

        </tr>

    <?php endforeach ?>

<?php endif ?>

</tbody>

</table>
```

surveys.view.php

```
<h4 class="mb-3">Add or deactivate surveys</h4>

<div class="row">

    <button id="addQuestion" class="btn blue lighten-1">

        <i class="material-icons">add</i>

    </button>

    <form action="/dashboard/add_survey" method="POST">

        <div id="addQuestionField" class="input-field col l6 right <?= empty($_GET['error']) ? 'hide' : '' ?>">

            <input id="addQuestionInput" name="add_question" type="text" />

            <label for="addQuestionInput">Ask them something</label>

            <button type="submit" class="btn blue lighten-1">

                <i class="material-icons">add</i>

            </button>

        </div>

    </form>

</div>

<table class="highlight striped centered">

    <thead>

        <tr>

            <td class="center">SurveyId</td>
```

```
<td class="center">Question</td>

<td class="center">IsActive</td>

</tr>

</thead>

<tbody>

<?php if(empty($surveys)) : ?>

<tr>

<td colspan="3">No surveys found!</td>

</tr>

<?php else : ?>

<?php foreach($surveys as $survey) : ?>

<?php $answers = $database->selectPollAnswers($survey->PollId) ?>

<tr>

<td><?= $survey->PollId ?></td>

<td>

<?= $survey->Question ?>

-

<?php foreach($answers as $answer) : ?>

<?= $answer->Answer . ' - ' . $answer->Result ?>

<?php endforeach ?>

</td>
```

```
<td>

    <?php if($survey->IsActive) : ?>

        <i class="white deactivate-survey material-icons green-text">check</i>

    <?php else : ?>

        <i class="material-icons red-text">close</i>

    <?php endif ?>

</td>

</tr>

<?php endforeach ?>

<?php endif ?>

</tbody>

</table>
```

users.view.php

```
<table class="highlight striped centered">

    <thead>

        <tr>

            <td class="center"></td>

            <td class="center">Username</td>

            <td class="center">Joined</td>

            <td class="center">Role</td>

        </tr>
```

```
</thead>

<tbody>

    <?php foreach($users as $user) : ?>

        <tr>

            <td></td>

            <td><?=$user->Username ?></td>

            <td><?=$user->CreatedAt ?></td>

            <td>

                <select name="role" class="role-change">

                    <?php foreach($roles as $role) : ?>

                        <option value="<?=$role->RoleId ?>" <?=$role->RoleId === $user->RoleId ? 'selected' : ''
?>><?=$role->RoleName ?></option>

                    <?php endforeach ?>

                </select>

            </td>

        </tr>

    <?php endforeach ?>

</tbody>

</table>
```

```
<?php

    $uri = substr(Request::getUri(), strpos(Request::getUri(), '/') + 1, strlen(Request::getUri()));

?>

<?php require 'admin/views/partials/main/head.php' ?>

<?php require 'admin/views/partials/main/sidenav.php' ?>

<div class="container">

    <nav class="mb-3">

        <div class="nav-wrapper blue lighten-1 pl-1">

            <div class="col s12">

                <a href="/dashboard/index" class="breadcrumb">Dashboard</a>

                <a href="/dashboard/<?= $uri ?>" class="breadcrumb page-name">

                    <?= $uri ?>

                </a>

                <span class="breadcrumb page-name">

                    <?= $message->MessageTitle ?>

                </span>

            </div>

        </div>

    </div>

</nav>

<a href="/dashboard/messages" class="btn blue">BACK</a>

<a href="/dashboard/messages?delete_message=<?= $_GET['view_message'] ?>" class="btn red">DELETE</a>
```

```
<h4>

    Title: <?= $message->MessageTitle ?>

</h4>

<p class="mb-3 pl-1">

    <?= $message->MessageBody ?>

</p>

<small>

    SentAt: <?= $message->CreatedAt ?>, by <?= $message->SenderMail ?>

</small>

</div>

<?php require 'admin/views/partials/main/footer.php' ?>
```

config

config.php

```
<?php

return

[

    'host' => 'xxxxx',

    'dbname' => 'xxxxx',

    'username' => 'xxxxx',

    'password' => 'xxxxx',

    'options' => [

        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION

    ]

];
```

main

index.php

```
<?php

$database = require 'core/bootstrap.php';

require Router::loadRoutes('routes/routes.php')

->direct(Request::getUri(), Request::getMethod());
```


routes

routes.php

```
<?php

$route->GET('index', 'controllers/index');

$route->GET('contact', 'controllers/contact');

$route->GET('author', 'controllers/author');

$route->GET('posts', 'controllers/posts/posts');

$route->GET('profile', 'controllers/user/profile');

$route->GET('survey', 'controllers/user/survey');

$route->GET('register', 'controllers/authentication/registration');

$route->GET('login', 'controllers/authentication/login');

$route->GET('logout', 'controllers/authentication/logout');

$route->GET('verify', 'controllers/validations/verify');

$route->GET('delete-post', 'controllers/posts/delete-post');

$route->GET('create-post', 'controllers/posts/create-post');

$route->GET('view-post', 'controllers/posts/view-post');
```

```
$router->GET('edit-post', 'controllers/posts/edit-post');

$router->GET('search', 'controllers/posts/search');

$router->GET('delete-comment', 'controllers/posts/delete-comment');


$router->POST('register', 'controllers/authentication/registration');

$router->POST('login', 'controllers/authentication/login');

$router->POST('create-post', 'controllers/posts/create-post');

$router->POST('edit-post', 'controllers/posts/edit-post');

$router->POST('update-image', 'controllers/user/update-image');

$router->POST('comment', 'controllers/posts/comment');

$router->POST('contact', 'controllers/contact');


$router->GET('dashboard/index', 'admin/controllers/index');

$router->GET('dashboard/admin', 'admin/controllers/profile');

$router->GET('dashboard/users', 'admin/controllers/users');

$router->GET('dashboard/posts', 'admin/controllers/posts');

$router->GET('dashboard/messages', 'admin/controllers/messages');

$router->GET('dashboard/surveys', 'admin/controllers/surveys');

$router->GET('dashboard/modify', 'admin/controllers/modify');

$router->POST('dashboard/modify', 'admin/controllers/modify');

$router->POST('dashboard/surveys', 'admin/controllers/surveys');
```

```
$router->POST('dashboard/add_survey', 'admin/controllers/add_survey');

$router->POST('dashboard/users', 'admin/controllers/users');

$router->POST('dashboard/posts', 'admin/controllers/posts');

$router->POST('surveys', 'controllers/surveys');
```

core

bootstrap.php

```
<?php

$configuration = require 'config/config.php';

require 'Router.php';

require 'vendor/autoload.php';

require 'Utilities.php';

require 'Request.php';

require 'database/Connection.php';

require 'database/QueryBuilder.php';

return new QueryBuilder(

    Connection::Connect($configuration)

);
```

Request.php

```
<?php

class Request

{

    public static function getUri()

    {

        $uri = trim($_SERVER['REQUEST_URI'], '/');

        if(strpos($uri, '?')) return substr($uri, 0, strpos($uri, '?'));

        return $uri;

    }

    public static function getMethod()

    {

        return $_SERVER['REQUEST_METHOD'];

    }

}
```

Router.php

```
<?php

class Router

{

    private $routes = [

        'GET' => [],

        'POST' => []

    ];

    public static function loadRoutes($file)

    {

        $router = new static;

        require $file;

        return $router;

    }

    public function GET($uri, $controller)

    {
```

```
        $this->routes['GET'][$uri] = $controller . '.php';

    }

    public function POST($uri, $controller)

    {

        $this->routes['POST'][$uri] = $controller . '.php';

    }

    public function direct($uri, $method)

    {

        if(array_key_exists($uri, $this->routes[$method]))

        {

            return $this->routes[$method][$uri];

        }

        else Utilities::redirect('/index');

    }

}
```

Utilities.php

```
<?php

class Utilities

{

    private $expressions = [

        'Password' => '/^[A-Z]{1}[a-z0-9!@#$$%^.&*]{7,19}$/',

        'Mail' => '/^[a-zA-Z0-9]([a-z]|[0-9])+\.?-?_?([a-z]|[0-9])*\.?([a-z]|[0-9])*@[a-z]{3,}\.([a-z]{2,4}\.)?([a-z]{2,4})$/',

        'FirstName' => '/^[A-Z]{1}[a-z]{2,14}$/',

        'LastName' => '/^[A-Z]{1}[a-z]{4,29}$/',

        'Username' => '/^([a-z]{1})[a-z0-9]{4,29}$/',

        'UserId' => '/^(([a-zA-Z0-9]([a-z]|[0-9])+\.?-?_?([a-z]|[0-9])*\.?([a-z]|[0-9])*@[a-z]{3,}\.([a-z]{2,4}\.)?([a-z]{2,4})))|((([a-z]{1})[a-z0-9]{4,29}))$/'

    ];

    public static function dieDump($variable)

    {

        die(var_dump($variable));

    }

}
```

```
public static function redirect($location)

{

    return header("location: {$location}");

}


public static function checkRegex($value, $name)

{

    $utility = new static;

    if(preg_match($utility->expressions[$name], $value)) return true;


    return false;

}

}
```


Connection.php

```
<?php

class Connection

{

    public static function Connect($cfg)

    {

        try

        {

            return new PDO

            (

                'mysql:host=' . $cfg['host'] . ';'.

                'dbname=' . $cfg['dbname'],

                $cfg['username'],

                $cfg['password']

            );

        }

        catch (PDOException $exception)

        {

            Utilities::dieDump($exception->getMessage());

        }

    }

}
```

QueryBuilder.php

```
<?php

class QueryBuilder

{

    private $pdo;

    public function __construct($pdo)

    {

        $this->pdo = $pdo;

    }

    public function selectRecords($table, $order = 0, $limit = 0, $from = 0, $approved = 0)

    {

        $query = sprintf(

            'SELECT * FROM %s %s %s %s',

            $table,

            $approved ? 'WHERE Approved = 1' : '',

            $order ? 'ORDER BY CreatedAt DESC' : '',

            $limit ? 'LIMIT ' . $from . ', 5' : ''

        );

    }

}
```

```
try

{

    $statement = $this->pdo->prepare($query);

    $statement->execute();

    return $statement->fetchAll(PDO::FETCH_CLASS);

}

catch (PDOException $exception)

{

    Utilities::dieDump($exception->getMessage());

}

}

public function selectLastRecords($table, $orderBy, $quantity, $approved = 0)

{

    $query = sprintf(

        'SELECT * FROM %s %s ORDER BY %s DESC LIMIT %s',

        $table,

        $approved ? 'WHERE Approved = 1' : '',

        $orderBy,
```

```
        $quantity

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute();

        return $statement->fetchAll(PDO::FETCH_CLASS);

    }

    catch (PDOException $exception)

    {

        Utilities::dieDump($exception->getMessage());

    }

}

public function selectFilteredRecord($table, $column, $condition)

{

    $query = sprintf(

        'SELECT * FROM %s WHERE %s = \'%s\' ',
```

```
        $table,  
  
        $column,  
  
        $condition  
    );  
  
    try  
    {  
  
        $statement = $this->pdo->prepare($query);  
  
        $statement->execute();  
  
        return $statement->fetch(PDO::FETCH_OBJ);  
    }  
  
    catch (PDOException $exception)  
    {  
  
        Utilities::dieDump($exception->getMessage());  
    }  
}  
  
public function selectFilteredRecords($table, $column, $condition)  
{
```

```
$query = sprintf(

    'SELECT * FROM %s WHERE %s = \'%s\' ',

    $table,

    $column,

    $condition

);

try

{

    $statement = $this->pdo->prepare($query);

    $statement->execute();

    return $statement->fetchAll(PDO::FETCH_CLASS);

}

catch (PDOException $exception)

{

    Utilities::dieDump($exception->getMessage());

}

}
```

```
public function selectCategoriesOfPost($post_id)

{

    $query = sprintf(

        'SELECT Categories.CategoryName

        FROM Categories

        INNER JOIN PostCategory ON Categories.CategoryId = PostCategory.CategoryId

        WHERE PostCategory.PostId = \'%s\'',

        $post_id,

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute();

        return $statement->fetchAll(PDO::FETCH_CLASS);

    }

    catch (PDOException $exception)

    {

        Utilities::dieDump($exception->getMessage());

    }

}
```

```
    }

}

public function selectPostsByCategory($categories)

{

    $query = sprintf(

        'SELECT DISTINCT Posts.PostId, Posts.PostTitle, Posts.PostBody, Posts.CreatedAt, Posts.UserId

        FROM Posts

        INNER JOIN PostCategory ON Posts.PostId = PostCategory.PostId

        WHERE PostCategory.CategoryId IN (%s) AND Posts.Approved = 1

        ORDER BY CreatedAt DESC',

        implode(' ', $categories),

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute();

        return $statement->fetchAll(PDO::FETCH_CLASS);

    }
```



```
        catch (PDOException $exception)

        {

            Utilities::dieDump($exception->getMessage());

        }

    }

    public function selectComments($postId)

    {

        $query = sprintf(

            'SELECT Users.UserId, CONCAT(Users.FirstName , " " , Users.LastName) AS \'FullName\', Users.UserName,

Users.ProfilePicture, Comments.CommentId, Comments.ParentId, Comments.CommentBody, Comments.PostId, Comments.CreatedAt

        FROM Users

        INNER JOIN Comments ON Users.UserId = Comments.UserId

        WHERE PostId = %s

        ORDER BY Comments.CreatedAt DESC',

        $postId

    );

    try

    {

        $statement = $this->pdo->prepare($query);
```

```
        $statement->execute();

        return $statement->fetchAll(PDO::FETCH_CLASS);
    }

    catch (PDOException $exception)
    {

        Utilities::dieDump($exception->getMessage());

    }

}

public function selectReplies($post_id, $parent_id)
{

    $query = sprintf(

        'SELECT Users.UserId, Users.FirstName, Users.LastName, Users.UserName, Users.ProfilePicture,
Comments.CommentId, Comments.ParentId, Comments.CommentBody, Comments.PostId, Comments.CreatedAt

        FROM Users INNER JOIN Comments ON Users.UserId = Comments.UserId

        WHERE Comments.PostId = %s AND Comments.ParentId = %s

        GROUP BY Comments.CommentId, Users.UserId, Users.FirstName, Users.LastName, Users.UserName,
Users.ProfilePicture, Comments.ParentId, Comments.CommentBody, Comments.PostId, Comments.CreatedAt',

        $post_id,
```

```
        $parent_id

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute();

        return $statement->fetchAll(PDO::FETCH_CLASS);

    }

    catch (PDOException $exception)

    {

        Utilities::dieDump($exception->getMessage());

    }

}

public function getRecordCount($table)

{

    $query = sprintf('SELECT COUNT(*) AS "Rows" FROM %s', $table);
```

```
try

{

    $statement = $this->pdo->prepare($query);

    $statement->execute();

    return $statement->fetch(PDO::FETCH_OBJ);

}

catch (PDOException $exception)

{

    Utilities::dieDump($exception->getMessage());

}

}

public function selectUser($id)

{

    $query = sprintf(

        'SELECT *

        FROM Users

        WHERE Username = \'%s\' OR UserMail = \'%s\' OR UserId = \'%s\' ',

        $id,

        $id,
```

```
        $id

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute();

        return $statement->fetch(PDO::FETCH_OBJ);

    }

    catch (PDOException $exception)

    {

    }

}

public function insertRecords($table, $parameters)

{

    $query = sprintf(

        'INSERT INTO %s (%s) VALUES (%s)',
```

```
        $table,

        implode(' ', array_keys($parameters)),

        ':' . implode(':', array_keys($parameters))

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute($parameters);

    }

    catch (PDOException $exception)

    {

        Utilities::dieDump($exception->getMessage());

    }

}

public function updateRecord($table, $column, $newValue, $condition, $conditionValue)

{

    $query = sprintf(

        'UPDATE %s SET %s = "%s" WHERE %s = \'%s\' ',
```

```
        $table,  
  
        $column,  
  
        $newValue,  
  
        $condition,  
  
        $conditionValue  
    );  
  
    try  
    {  
  
        $statement = $this->pdo->prepare($query);  
  
        $statement->execute();  
    }  
  
    catch (PDOException $exception)  
    {  
  
        Utilities::dieDump($exception->getMessage());  
    }  
  
}
```



```
public function updateAnswer($answer, $poll_id, $user_id)
```

```
{

    $query = sprintf(

        'UPDATE PollAnswers SET Answer = \'%s\' WHERE PollId = \'%s\' AND UserId = \'%s\' ',

        $answer,

        $poll_id,

        $user_id

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute();

    }

    catch (PDOException $exception)

    {

        Utilities::dieDump($exception->getMessage());

    }

}

public function updatePostBody($new_value, $condition_value)
```



```
{  
  
    $query = sprintf(  
  
        'UPDATE Posts SET PostBody = :body WHERE PostId = "%s" ',  
  
        $condition_value  
  
    );  
  
}
```

```
try
```

```
{
```

```
    $statement = $this->pdo->prepare($query);  
  
    $statement->bindParam(':body', $new_value);  
  
    $statement->execute();  
  
}
```

```
catch (PDOException $exception)
```

```
{
```

```
    Utilities::dieDump($exception->getMessage());  
  
}
```

```
}
```

```
public function deleteRecords($table, $column, $value)
```

```
{
```

```
    $query = sprintf(  
  
        'DELETE FROM %s WHERE %s = %s',  
  
        $table, $column, $value
```

```
        $table,  
  
        $column,  
  
        $value  
    );  
  
    try  
    {  
  
        $statement = $this->pdo->prepare($query);  
  
        $statement->execute();  
    }  
  
    catch (PDOException $exception)  
    {  
  
        Utilities::dieDump($exception->getMessage());  
    }  
}  
  
public function deleteAllRecords($table)  
{  
  
    $query = sprintf(  
  
        'DELETE FROM %s',  
  
        $table  
    );  
}
```

```
try

{

    $statement = $this->pdo->prepare($query);

    $statement->execute();

}

catch (PDOException $exception)

{

    Utilities::dieDump($exception->getMessage());

}

}

public function selectPolls($user_id)

{

    $query = sprintf(

        'SELECT *

        FROM Poll

        LEFT JOIN PollAnswers ON Poll.PollId = PollAnswers.PollId

        WHERE (PollAnswers.UserId = \'%s\' AND PollAnswers.Answer IS NULL) AND Poll.IsActive = 1',

        $user_id

    );
```

```
try

{

    $statement = $this->pdo->prepare($query);

    $statement->execute();

    return $statement->fetchAll(PDO::FETCH_CLASS);

}

catch (PDOException $exception)

{

    Utilities::dieDump($exception->getMessage());

}

}

public function selectPollAnswers($poll_id)

{

    $query = sprintf(

        'SELECT Answer, Count(UserId) AS Result

        FROM PollAnswers

        WHERE PollId = %s AND Answer IS NOT NULL
```

```
        GROUP BY Answer',

        $poll_id

    );

    try

    {

        $statement = $this->pdo->prepare($query);

        $statement->execute();

        return $statement->fetchAll(PDO::FETCH_CLASS);

    }

    catch (PDOException $exception)

    {

        Utilities::dieDump($exception->getMessage());

    }

}
```

views/authentication

login.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container mb-5">

    <h1>Log In</h1>

</div>

<div class="container mb-5">

    <div class="container">

        <div class="container">

            <form action="/login" method="POST" id="loginForm">

                <div class="input-field mb-3">

                    <input id="userId" placeholder="Type here ..." name="user_id" type="text" value="<?=
!empty($errors['password']) ? $values['credentials'] : '' ?>" />

                    <label for="userId">Username or E-mail</label>

                    <?php if(isset($errors['credentials'])) : ?>

                        <span class="red-text">

                            <?= $errors['credentials'] ?>

                        </span>
```

```
<?php else : ?>

    <span></span>

<?php endif ?>

</div>

<div class="input-field mb-1">

    <input id="userPassword" placeholder="Enter your password ..." name="user_password"

type="password" />

    <label for="userPassword">Password</label>

    <?php if(isset($errors['password'])) : ?>

        <span class="red-text">

            <?= $errors['password'] ?>

        </span>

    <?php else : ?>

        <span></span>

    <?php endif ?>

</div>

<div class="row mb-3 center">

    <button class="btn btn-large blue lighten-1 waves-effect waves-light" name="login_user"

id="loginButton">Log In</button>

    </div>

<div class="row center">

    Don't have an account? <a href="/register" class="link">Register</a>
```

```
        </div>

    </form>

</div>

</div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

registration.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container mb-5">

    <h1>Register</h1>

</div>

<div class="container mb-5">

    <div class="container">

        <div class="container">

            <form action="/register" method="POST" id="registrationForm">

                <div class="input-field mb-3">

                    <input id="firstName" placeholder="e.g. John" name="first_name" type="text" value="<?=

isset($values['firstname']) ? $values['firstname'] : '' ?>" />
```



```
<label for="firstName">First name</label>

<?php if(isset($errors['firstname'])) : ?>

    <span class="red-text">

        <?= $errors['firstname'] ?>

    </span>

<?php else : ?>

    <span></span>

<?php endif ?>

</div>

<div class="input-field mb-3">

    <input id="lastName" placeholder="e.g. Doe" name="last_name" type="text" value="<?=
isset($values['lastname']) ? $values['lastname'] : '' ?>" />

    <label for="lastName">Last name</label>

    <?php if(isset($errors['lastname'])) : ?>

        <span class="red-text">

            <?= $errors['lastname'] ?>

        </span>

    <?php else : ?>

        <span></span>

    <?php endif ?>

</div>

<div class="input-field mb-3">
```

```
<input id="userName" placeholder="e.g. johndoe" name="user_name" type="text" value="<?=
isset($values['username']) ? $values['username'] : '' ?>" />

<label for="userName">Username</label>

<?php if(isset($errors['username'])) : ?>

    <span class="red-text">

        <?= $errors['username'] ?>

    </span>

<?php else : ?>

    <span></span>

<?php endif ?>

</div>

<div class="input-field mb-3">

    <input id="userMail" placeholder="e.g. johndoe@example.com" name="user_mail" type="email"
value="<?= isset($values['usermail']) ? $values['usermail'] : '' ?>" />

    <label for="userMail">E-Mail</label>

    <?php if(isset($errors['usermail'])) : ?>

        <span class="red-text">

            <?= $errors['usermail'] ?>

        </span>

    <?php else : ?>

        <span></span>

    <?php endif ?>
```

```
</div>

<div class="input-field mb-3">

    <input id="password" placeholder="Min. 8 chars, one capital, one symbol! (max. - 20)"
name="password" type="password" />

    <label for="password">Password</label>

    <?php if(isset($errors['password'])) : ?>

        <span class="red-text">

            <?= $errors['password'] ?>

        </span>

    <?php else : ?>

        <span></span>

    <?php endif ?>

</div>

<div class="input-field mb-3">

    <input id="rePassword" placeholder="Repeat password" name="re_password" type="password" />

    <label for="rePassword">Re-enter password</label>

    <?php if(isset($errors['re_password'])) : ?>

        <span class="red-text">

            <?= $errors['re_password'] ?>

        </span>

    <?php else : ?>

        <span></span>
```

```

        <?php endif ?>

    </div>

    <div class="row mb-3 center">

        <button class="btn btn-large blue lighten-1 waves-effect waves-light" name="register_user"
id="registerButton">REGISTER</button>

    </div>

    <div class="row center">

        Already have an account? <a href="/login" class="link">Log In</a>

    </div>

</form>

</div>

</div>

</div>

<?php require 'resources/views/partials/footer.php' ?>

```

verify.view.php

```

<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container">

    <div class="container verify-container center">

        <?php if(isset($token)) : ?>

```

```
<h3>Thank you for your registration!</h3>

<h4>Your account has been validated!</h4>

<?php else : ?>



<h3>Thank you for your registration!</h3>

<h4>Please, check your email for the verification token!</h4>

<?php endif ?>

</div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

views/pages

author.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<main>

<div class="parallax-container container2">

<div class="parallax">

&ltimg src="public/img/cover.jpg" alt="Matrix Binary rain"/>

</div>

<div class="container">

<h1 class="white-text center">author</h1>
```

```
</div>

</div>

<section class="section blue lighten-7 p-0">

    <div class="cross-section"></div>

    <div class="container">

        <div class="row">

            <div class="col s12 m6 l4 offset-l2">

            </div>

            <div class="flow-text col s12 m6 l6">

                <blockquote class="quote-border">Hello,</blockquote>

                my name is Lazar Lalović. I am 22 years old. I was born in Zvornik and raised in a small town

called Milići,

                Serbian Republic. Now, I am a second-year student at: "College of Vocational Studies for

Information and Communication Technologies",

                where I am majoring in Web programming.

                <br />

                <br />
```

```

        <a target="_blank" href="https://laxon17.github.io/portfolio" class="btn btn-large blue lighten-1
waves-effect waves-light">More about me</a>

    </div>

</div>

</div>

<div class="cross-section opposite"></div>

</section>

</main>

<?php require 'resources/views/partials/footer.php' ?>

```

contact.view.php

```

<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<?php

    $user = $database->selectFilteredRecord('Users', 'UserId', $_SESSION['user_id']);

?>

<div class="container mb-5">

    <h2>

```

```

        <?= 'Hello, write us a message!' ?>

    </h2>

    <?php if(isset($_GET['success'])) : ?>

        <span class="green lighten-5 p-1 green-text text-darken-4 center">

            <?= $_GET['success'] ?>

        </span>

    <?php endif ?>

</div>

<div class="container">

    <div class="container">

        <div class="container center">

            <form action="/contact" method="POST" id="loginForm">

                <div class="row">

                    <div class="col 16 pl-0">

                        <div class="input-field mb-2">

                            <input id="contactMail" name="contact_mail" type="email" value="<?= isset($user) ? $user->UserMail : '' ?>" required />

                            <label for="contactMail">E-Mail</label>

                            <?php if(isset($errors['credentials'])) : ?>

                                <span class="red-text">

                                    <?= $errors['credentials'] ?>

```



```
        </span>

        <?php else : ?>

        <span></span>

        <?php endif ?>

    </div>

</div>

<div class="col 16 pr-0">

    <div class="input-field mb-2">

        <input id="messageTitle" name="message_title" type="text" />

        <label for="messageTitle">Message Title</label>

    </div>

</div>

</div>

<div class="input-field mb-1">

    <textarea name="message_body" class="materialize-textarea" id="messageBody" required></textarea>

    <label for="messageBody">Write us something...</label>

    <?php if(isset($errors['message'])) : ?>

        <span class="red-text">

            <?= $errors['message'] ?>

        </span>

    <?php else : ?>

        <span></span>
```

```
        <?php endif ?>

    </div>

    <div class="row mb-3 center">

        <button class="btn btn-large blue lighten-1 waves-effect waves-light" name="send_message"
id="sendButton">SEND</button>

    </div>

</form>

</div>

</div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

index.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="parallax-container">

    <div class="parallax">

    </div>

    <div class="container center">

        <h1 class="white-text">
```

```
<?= empty($user_info->FirstName) ? 'Welcome!' : 'Hello ' . $user_info->FirstName . '!' How are you
today?'?>

</h1>

</div>

</div>

<div class="section white p-5">

<div class="row container center">

<h2 class="header">Welcome to WireFrame</h2>

<p class="grey-text text-darken-3 lighten-3">

    The Developer's Blog to Keep You at the Top of Your Game.

</p>

</div>

</div>

<div class="parallax-container">

<div class="parallax">



</div>

</div>

<div class="section white p-5">

<div class="row container">

<h2 class="header">Stay informed!</h2>

<p class="grey-text text-darken-3 lighten-3">
```

The field of software engineering is constantly developing. (ba dum tss) With tech, frameworks, and programming best practices changing every day—even the best developers risk falling behind.

Nobody really wants to dedicate large chunks of their time to continued education. We get that. Fortunately, the internet is littered with developer blogs. And, while we here at WireFrame are anti-littering, we are definitely pro-innovation, pro-growth and pro-learning from the trials and errors of others so we don't have to waste time ourselves.

</p>

</div>

</div>

<div class="parallax-container">

<div class="parallax">

</div>

</div>

<div class="section white p-5">

<div class="row container">

<h2 class="header">Interested in writing?</h2>

<p class="grey-text text-darken-3 lighten-3">

Simply create a profile and start writing programming topics, to help future generations!


```
Join us, and we will change and create the web together!

<br />

<br />

<a href="/register" class="btn btn-medium blue lighten-1 waves-effect waves-light">Register</a>

</p>

</div>

</div>

<div class="parallax-container">

  <div class="parallax">

  </div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

posts.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container mb-5">

  <h1>Posts</h1>

</div>

<div class="container center">

  <div class="container center">
```

```

        <div class="row center">

            <div class="input-field col s12 m11 offset-m1 l11 offset-l1">

                <i class="material-icons prefix">search</i>

                <input type="text" id="searchPosts" name="search_posts" />

                <label for="searchPosts">Search for post</label>

            </div>

        </div>

    </div>

</div>

<div class="row">

    <div class="col s12 m2 offset-m1 l2 offset-l1 mb-3">

        <p class="flow-text mb-1">Categories</p>

        <form action="/posts" method="GET">

            <ul class="mb-3">

                <?php foreach($categories as $category) : ?>

                    <li>

                        <label>

                            <input type="checkbox" name="categories[]" value="<?= $category->CategoryId ?>" <?=
in_array($category->CategoryId, $chosen_categories) ? 'checked' : '' ?>/>

                            <span><?= $category->CategoryName ?></span>

                        </label>

                    </li>

```

```
<?php endforeach ?>

</ul>

<button type="submit" class="btn btn-small blue lighten-1 waves-effect waves-light">FILTER</button>

</form>

</div>

<div class="col s12 m2 push-m7 l2 push-l7 mb-3">

<p class="flow-text mb-1">Latest</p>

<ul>

<?php if(!empty($posts)) : ?>

<?php foreach($recent_posts as $recent_post) : ?>

<li class="recent-post">

<a href="/view-post?id=<?= $recent_post->PostId ?>">

<?= $recent_post->PostTitle ?>

</a>

<br/>

<p>

<?= $recent_post->CreatedAt ?>

</p>

</li>

<?php endforeach ?>

<?php else : ?>

<p class="grey-text">No posts added...</p>
```

```
<?php endif ?>

</ul>

</div>

<div class="col s12 m7 pull-m2 l7 pull-l2">

    <table class="striped highlight mb-3">

        <tbody id="postContainer">

            <?php if(empty($posts)) : ?>

                <tr>

                    <h5 class="grey-text text-lighten-1">No posts available at this moment!</h5>

                </tr>

            <?php else : ?>

                <?php foreach($posts as $post) : ?>

                    <tr>

                        <td class="post-list">

                            <a href="/view-post?id=<?= $post->PostId ?>">

                                <?= $post->PostTitle ?>

                            </a>

                        </td>

                    </tr>

                <?php endforeach ?>

            <?php endif ?>

        </tbody>
```



```
</table>

<?php if(!empty($posts) || count($posts) < 10) : ?>

    <ul class="pagination">

        <?php for($i = 1; $i <= $pages_number; $i++) : ?>

            <?php

                if(empty($_GET['page'])) $page = 1;

                else $page = $_GET['page'];

            ?>

            <li class="<?=$page == $i ? 'active blue lighten-1' : '' ?>">

                <a href="/posts?page=<?=$i ?>">

                    <?=$i ?>

                </a>

            </li>

        <?php endfor ?>

    </ul>

<?php endif ?>

</div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

views/posts

create-post.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container mb-3">

    <h2>

        <?= 'Create post' ?>

    </h2>

</div>

<div class="container">

    <div class="container">

        <div class="container">

            <form action="/create-post" method="POST" enctype="multipart/form-data">

                <div class="input-field mb-3">

                    <input type="text" placeholder="CSS Grid system..." id="postTitle" value="<?= $values['title'] ?>"

name="post_title" required />

                    <label for="postTitle">Title</label>

                    <?php if(isset($errors['title'])) : ?>

                        <span class="red-text"><?= $errors['title'] ?></span>

                    <?php else : ?>
```

```

        <span></span>

        <?php endif ?>

    </div>

    <div class="input-field mb-3">

        <select id="categorySelect" name="category_select[]" multiple>

            <option value="0" disabled selected>Categories:</option>

            <?php foreach($categories as $category) : ?>

                <option class="blue-text" value="<?= $category->CategoryId ?>"><?= $category-
>CategoryName ?></option>

            <?php endforeach ?>

        </select>

        <label for="categorySelect">Category</label>

        <?php if(isset($errors['category'])) : ?>

            <span class="red-text"><?= $errors['category'] ?></span>

        <?php else : ?>

            <span></span>

        <?php endif ?>

    </div>

    <div class="input-field">

        <textarea id="textarea1" name="post_body" placeholder="Write something..." class="materialize-
textarea" required><?= $values['body'] ?></textarea>

        <label for="textarea1">Post body</label>

```

```
<?php if(isset($errors['body'])) : ?>

    <span class="red-text"><?= $errors['body'] ?></span>

<?php else : ?>

    <span></span>

<?php endif ?>

</div>

<div class="row">

    <div class="file-field input-field">

        <div class="btn blue lighten-1 mr-1">

            <span>Image</span>

            <input type="file" name="image_path" id="coverImage" accept="image/png, image/jpg,
image/jpeg" />

        </div>

        <div class="file-path-wrapper">

            <input class="file-path" type="text">

        </div>

        <p class="pt-1">Upload cover image &#40;important&#41;</p>

        <?php if(isset($errors['image'])) : ?>

            <span class="red-text">

                <?= $errors['image'] ?>

            </span>

        <?php else : ?>
```

```
                <span></span>

                <?php endif ?>

            </div>

        </div>

        <div class="row center">

            <button type="submit" class="btn btn-large blue lighten-1">CREATE POST</button>

        </div>

    </form>

</div>

</div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

edit-post.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container mb-3">

    <h2>

        <?= 'Edit post' ?>

    </h2>

</div>
```

```

<div class="container ms">

    <div class="container">

        <div class="container">

            <form action="/edit-post" method="POST" enctype="multipart/form-data">

                <div class="input-field mb-3">

                    <input type="text" placeholder="CSS Grid system..." id="postTitle" value="<?= $values['title'] ?>"

name="post_title" required />

                    <label for="postTitle">Title</label>

                    <?php if(isset($errors['title'])) : ?>

                        <span class="red-text"><?= $errors['title'] ?></span>

                    <?php else : ?>

                        <span></span>

                    <?php endif ?>

                </div>

                <div class="input-field mb-3">

                    <select id="categorySelect" name="category_select[]" multiple>

                        <option value="0" disabled selected>Categories:</option>

                        <?php foreach($categories as $category) : ?>

                            <option class="blue-text" value="<?= $category->CategoryId ?>" <?= in_array($category->CategoryId, $post_categories) ? 'selected' : '' ?> ><?= $category->CategoryName ?></option>

                        <?php endforeach ?>

                    </select>

```

```
<label for="categorySelect">Category</label>

<?php if(isset($errors['category'])) : ?>

    <span class="red-text"><?= $errors['category'] ?></span>

<?php else : ?>

    <span></span>

<?php endif ?>

</div>

<div class="input-field">

    <textarea id="textarea1" name="post_body" placeholder="Write something..." class="materialize-
textarea" required><?= $values['body'] ?></textarea>

    <label for="textarea1">Post body</label>

    <?php if(isset($errors['body'])) : ?>

        <span class="red-text"><?= $errors['body'] ?></span>

    <?php else : ?>

        <span></span>

    <?php endif ?>

</div>

<div class="row">

    <div class="file-field input-field">

        <div class="btn blue lighten-1 mr-1">

            <span>Image</span>
```

```

        <input type="file" name="new_path" id="coverImage" accept="image/png, image/jpg,
image/jpeg" />

    </div>

    <div class="file-path-wrapper">

        <input class="file-path" type="text" value="<?= $values['path'] ?>" />

        <input type="hidden" value="<?= $values['path'] ?>" name="old_path" />

    </div>

    <p class="pt-1">Upload cover image &#40;important&#41;</p>

    <?php if(isset($errors['image'])) : ?>

        <span class="red-text">

            <?= $errors['image'] ?>

        </span>

    <?php else : ?>

        <span></span>

    <?php endif ?>

</div>

</div>

<div class="row center">

    <button type="submit" class="btn btn-large blue lighten-1">SAVE POST</button>

</div>

<input type="hidden" name="post_id" value="<?= $_GET['id'] ?>" />

</form>

```



```

        </div>

    </div>

</div>

<?php require 'resources/views/partials/footer.php' ?>

```

view-post.view.php

```

<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container mb-3">

    <h2>

        <?= $post->PostTitle ?>

    </h2>

    <p class="mb-1">

        <?= 'Written by: ' . '<u>' . $writer->FirstName . ' ' . $writer->LastName . '</u>' . ' at: ' . $post-
>CreatedAt ?>

    </p>

    <div class="row">

        <?php if($_SESSION['user_id'] === $post->UserId || $_SESSION['role'] === 1) : ?>

            <a class="btn btn-small red waves-effect" href="/delete-post?id=<?= $post->PostId ?>">DELETE POST</a>

        <?php if($_SESSION['role'] !== 1 || $_SESSION['user_id'] === $post->UserId) : ?>

            <a class="btn btn-small yellow darken-2 waves-effect" href="/edit-post?id=<?= $post->PostId ?>">EDIT
POST</a>

```

```
<?php endif ?>

<?php endif ?>

</div>

</div>

<div class="container mb-3">

    <div class="container">

        <p class="mb-3">

            <?= $post->PostBody ?>

        </p>

        <blockquote class="mb-3">

            Tags:

            <p class="grey-text text-darken-1">

                <?php foreach($categories as $category) : ?>

                    #<?= $category->CategoryName ?>

                <?php endforeach ?>

            </p>

        </blockquote>

    </div>

</div>

<div class="divider"></div>
```

```
<div class="container">

    <div class="container">

        <h3 class="mb-3">Comments</h3>

        <?php require 'resources/views/partials/post-comment.php' ?>

    </div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

views/users

profile.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container center pt-5 mb-3">

    <h4 class="responsive-text">

        <?php if(isset($_SESSION['user_id']) && $_SESSION['user_id'] === $user->UserId) : ?>

            <?= 'Hello, ' . $user->FirstName . '!' ?>

        <?php else : ?>

            <?= $user->FirstName . ' ' . $user->LastName ?>

        <?php endif ?>

    </h4>

</div>
```

```

<div class="container">

    <div class="row" id="userInfo">

        <div class="col s10 m10 l2 offset-s1 offset-m1 offset-l1 pt-2" id="profilePictureContainer">

            <div id="profileImage">

            </div>

        </div>

    </div>

    <div class="col s12 m10 l6 offset-m1 offset-l2 mb-3" id="userBio">

        <b>Username:</b> <?= $user->Username ?> <br><br>

        <b>Mail:</b> <?= $user->UserMail ?> <br><br>

        <b>Joined:</b> <?= $user->CreatedAt ?> <br><br>

        <b>Role: </b> <?= $role_name->RoleName ?> <br><br>

        <?php if(isset($_SESSION['user_id']) && $_SESSION['user_id'] === $user->UserId) : ?>

            <b>Change profile picture:</b>

            <form action="/update-image" method="POST" enctype="multipart/form-data">

                <div class="row">

                    <div class="col l8">

                        <div class="file-field input-field">

                            <div class="btn btn-small blue lighten-1">

                                <span>UPLOAD</span>

```

```

        <input type="file" name="image_path" id="profileImage" accept="image/png,
image/jpg, image/jpeg" />

    </div>

    <div class="file-path-wrapper">

        <input class="file-path" type="text" />

    </div>

</div>

</div>

<div class="col 12 pt-1">

    <button class="btn btn-medium blue lighten-1 waves-effect" type="submit" >UPDATE</button>

</div>

</div>

</form>

<a href="/survey?user_id=<?= $_SESSION['user_id'] ?>" class="btn blue lighten-1">Take poll</a>

<?php endif ?>

</div>

</div>

<div class="row mb-3">

    <?php if($_SESSION['user_id'] === $user->UserId) : ?>

        <div class="col center mb-1 push-11">

```

```

        <a class="btn btn-large waves-effect waves-light blue lighten-1 <?= empty($posts) ? 'pulse' : '' ?>"
href="/create-post">

        <i class="material-icons">add</i>

    </a>

</div>

<?php endif ?>

<div class="col s12 <?= $_SESSION['user_id'] === $user->UserId ? '19 offset-11' : '18 offset-12' ?>">

    <h5 class="responsive-text"><?= $_SESSION['user_id'] === $user->UserId ? 'My' : $user->FirstName . '\s'
?> posts</h5>

    <div class="divider mb-3"></div>

    <?php if(empty($posts)) : ?>

        <?php if($_SESSION['user_id'] === $user->UserId) : ?>

            <p class="p1-1">You haven't posted anything yet!</p>

        <?php else : ?>

            <p class="p1-1"><?= $user->FirstName ?> hasn't posted anything yet!</p>

        <?php endif ?>

    <?php else : ?>

        <table class="striped highlight centered">

            <thead>

                <tr>

                    <td class="center">PostId</td>

                    <td class="center">Post title</td>

```

```
<?php if($_SESSION['user_id'] === $user->UserId) : ?>

    <td class="center">Edit</td>

    <td class="center">Remove</td>

<?php endif ?>

</tr>

</thead>

<tbody>

<?php foreach($posts as $index => $post) : ?>

    <tr>

        <td><?= ++$index ?></td>

        <td>

            <a href="/view-post?id=<?= $post->PostId ?>"><?= $post->PostTitle ?></a>

        </td>

        <?php if($_SESSION['user_id'] === $user->UserId) : ?>

            <td>

                <a href="/edit-post?id=<?= $post->PostId ?>">

                    <i class="material-icons yellow-text">edit</i>

                </a>

            </td>

            <td>

                <a href="/delete-post?id=<?= $post->PostId ?>">

                    <i class="material-icons red-text">cancel</i>

                </a>

            </td>

        </?php ?>

    </tr>

</?php ?>
```

```
                </a>

            </td>

        <?php endif ?>

    </tr>

    <?php endforeach ?>

</tbody>

</table>

<?php endif ?>

</div>

</div>

</div>

<?php require 'resources/views/partials/footer.php' ?>
```

survey.view.php

```
<?php require 'resources/views/partials/head.php' ?>

<?php require 'resources/views/partials/navigation.php' ?>

<div class="container ms">

    <h3 class="mb-3">Take our surveys</h3>

    <table class="highlight striped" >

        <thead>

            <tr>

                <td class="center">PollId</td>
```



```

        <td class="center">Question</td>

        <td class="center">Your answer</td>

    </tr>

</thead>

<tbody>

    <?php if(!empty($polls)) : ?>

        <?php foreach($polls as $poll) : ?>

            <tr>

                <td class="center"><?= $poll->PollId ?></td>

                <td class="center"><?= $poll->Question ?></td>

                <td>

                    <form action="/surveys" method="post">

                        <input type="hidden" name="user_id" value="<?= $_SESSION['user_id'] ?>" />

                        <input type="hidden" name="poll_id" value="<?= $poll->PollId ?>" />

                        <p>

                            <label>

                                <input name="poll_answer" type="radio" value="Never" />

                                <span class="black-text">Never</span>

                            </label>

                        </p>

                    ><p>

                        <label>

```

```

        <input name="poll_answer" type="radio" value="Sometimes" />

        <span class="black-text">Sometimes</span>

    </label>

</p>

<p class="mb-1">

    <label>

        <input name="poll_answer" type="radio" value="Always" />

        <span class="black-text">Always</span>

    </label>

</p>

<button type="submit" class="btn btn-small blue lighten-1">SEND</button>

</form>

</td>

</tr>

<?php endforeach ?>

<?php else : ?>

    <tr>

        <td class="center" colspan="3">No polls active!</td>

    </tr>

<?php endif ?>

</tbody>

</table>

```

```
</div>

<?php require 'resources/views/partials/footer.php' ?>
```

views/partials

head.php

```
<?php session_start() ?>

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <meta name="keywords" content="Blog, programming, topics, developers, front end, back end, php, c#, sql,
databases, mysql, javascript, .net" />

    <meta name="description" content="Hello, we are WireFrame! Sign up and help us grow in the development
community!" />

    <meta name="author" content="Lazar Lalovic 62/20" />

    <link rel="preconnect" href="https://fonts.googleapis.com"/>

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
```

```

        <link href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,400;1,200&display=swap"
rel="stylesheet" />

        <!--Import Google Icon Font-->

        <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />

        <link href="public/img/icon.png" rel="icon" />

        <!-- Compiled and minified CSS -->

        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"
/>

        <link rel="stylesheet" href="public/css/style.css" />

        <link rel="stylesheet" href="public/css/responsive.css" />

        <script type="text/javascript" src="public/js/main.js" defer></script>

        <script src="node_modules/ckeditor4/ckeditor.js"></script>

        <title>Wireframe | <?= $page_title; ?></title>

    </head>

    <body>

```

footer.php

```

<?php

    $links = $database->selectRecords('Navigation');

?>

    <div class="preloader white">

```

```
<div class="preloader-wrapper active big">

  <div class="spinner-layer spinner-blue-only">

    <div class="circle-clipper left">

      <div class="circle"></div>

    </div>

    <div class="gap-patch">

      <div class="circle"></div>

    </div>

    <div class="circle-clipper right">

      <div class="circle"></div>

    </div>

  </div>

</div>

</div>

<?php if(Request::getUri() !== 'dashboard') : ?>

  <footer class="page-footer grey darken-3">

    <div class="container">

      <div class="row">

        <div class="col s12 m3 l3 left-on-med-and-up center-on-small-only">

          <h2 class="white-text flow-text">USEFUL LINKS</h2>

          <ul class="blog-info">

            <?php foreach($links as $index => $link) : ?>
```

```
<?php if($index < 4 || ($index > 6 && $index < 9)) : ?>

    <li>

        <a href="<?= $link->LinkPath ?>"><?= $link->LinkName ?></a>

    </li>

<?php endif ?>

<?php endforeach ?>

</ul>

</div>

<div class="col s12 m6 l6 center-align">

    <div class="row">

        <h2 class="flow-text">CONNECT WITH US</h2>

        <a href="resources/assets/feed.rss"></a>

        <a target="_blank" href="https://www.facebook.com"></a>

        <a target="_blank" href="https://www.twitter.com"></a>

        <a target="_blank" href="https://www.linkedin.com"></a>

        <a target="_blank" href="https://www.github.com/laxon17"></a>

    </div>
```

```
<div class="row">

  <div class="col s6 m6 l6 offset-s3 offset-m3 offset-l3">

    <h3 class="grey-text">WireFrame</h3>

  </div>

</div>

</div>

<div class="col s12 m3 l3 left-on-med-and-up center-on-small-only">

  <h2 class="white-text flow-text">SITE ASSETS</h2>

  <ul class="blog-info">

    <li>

      <a class="grey-text text-lighten-4" target="_blank"
href="resources/assets/documentation.pdf">Documentation</a>

    </li>

    <li><a class="grey-text text-lighten-4" target="_blank"
href="resources/assets/sitemap.xml">Sitemap</a></li>

    <li><a class="grey-text text-lighten-4" target="_blank"
href="public/js/main.js">Script.js</a></li>

  </ul>

</div>

</div>

</div>

<div class="footer-copyright grey darken-4">
```

```
<div class="container center">

    Copyright &copy; WireFrame 2022. All Rights Reserved.

</div>

</div>

</footer>

<?php endif ?>

<!-- jQuery -->

<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-
/xUj+30JU5yEx1q6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>

<!-- Compiled and minified JavaScript -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>

<script>

    // Replace the <textarea id="editor1"> with a CKEditor 4

    // instance, using default configuration.

    CKEDITOR.replace('textarea1')

</script>

</body>

</html>
```


navigation.php

```
<?php

$links = $database->selectRecords('Navigation');

$position = '/' . Request::getUri();

if(isset($_SESSION['user_id'])) $user_info = $database->selectUser($_SESSION['user_id']);

?>

<div class="navbar-fixed">

    <nav class="blue lighten-1">

        <div class="nav-wrapper container">

            <a href="<?= $links[0]->LinkPath ?>" class="brand-logo center">WireFrame</a>

            <a data-target="hiddenNavigation" class="sidenav-trigger">

                <i class="material-icons white-text">menu</i>

            </a>

            <ul id="visibleNavigation" class="left hide-on-med-and-down">

                <?php for($i = 1; $i < 4; $i++) : ?>

                    <li class="<?= $position === $links[$i]->LinkPath ? 'active' : '' ?>">
```

```

        <a href="<?= $links[$i]->LinkPath ?>"><?= $links[$i]->LinkName ?></a>

    </li>

<?php endfor ?>

</ul>

<ul class="right hide-on-med-and-down">

    <?php if(isset($_SESSION['user_id']) && $_SESSION['role'] === 1) : ?>

        <li>

            <a href="<?= $links[4]->LinkPath ?>"><?= $links[4]->LinkName ?></a>

        </li>

    <?php endif ?>

    <?php if(isset($_SESSION['user_id'])) : ?>

        <li class="<?= $position === $links[5]->LinkPath && $_SESSION['username'] === $_GET['username'] ? 'active' : ''
?>">

            <a href="<?= $links[5]->LinkPath . '?username=' . $_SESSION['username'] ?>"><?= $links[5]->LinkName ?></a>

        </li>

        <li class="<?= $position === $links[6]->LinkPath ? 'active' : '' ?>">

            <a href="<?= $links[6]->LinkPath ?>"><?= $links[6]->LinkName ?></a>

        </li>

    <?php endif ?>

    <?php if(empty($_SESSION['user_id'])) : ?>

        <?php for($i = 7; $i < 9; $i++) : ?>

```

```
<li class="<?=$position === $links[$i]->LinkPath ? 'active' : '' ?>">

    <a href="<?=$links[$i]->LinkPath ?>">

        <?=$links[$i]->LinkName ?>

    </a>

</li>

<?php endfor ?>

<?php endif ?>

</ul>

</div>

</nav>

</div>

<ul id="hiddenNavigation" class="sidenav">

    <?php if(isset($_SESSION['user_id'])) : ?>

        <li>

            <div class="user-view">

                <div class="background">

                </div>

                <a href="#user"></a>

                <a href="#name"><span class="white-text name"><?=$user_info->FirstName . ' ' . $user_info->LastName ?></span></a>

                <a href="#email"><span class="white-text email"><?=$user_info->UserMail ?></span></a>
```

```
</div>

</li>

<?php endif ?>

<?php for($i = 1; $i < 4; $i++) : ?>

    <li class="<?=$position === $links[$i]->LinkPath ? 'active' : '' ?>">

        <a href="<?=$links[$i]->LinkPath ?>"><?=$links[$i]->LinkName ?></a>

    </li>

<?php endfor ?>

<?php if(isset($_SESSION['user_id']) && $_SESSION['role'] === 1) : ?>

    <li>

        <a href="<?=$links[4]->LinkPath ?>"><?=$links[4]->LinkName ?></a>

    </li>

<?php endif ?>

<?php if(isset($_SESSION['user_id'])) : ?>

    <li class="<?=$position === $links[5]->LinkPath && $_SESSION['username'] === $_GET['username'] ? 'active' : '' ?>">

        <a href="<?=$links[5]->LinkPath . '?username=' . $_SESSION['username'] ?>"><?=$links[5]->LinkName ?></a>

    </li>

    <li class="<?=$position === $links[6]->LinkPath ? 'active' : '' ?>">

        <a href="<?=$links[6]->LinkPath ?>"><?=$links[6]->LinkName ?></a>

    </li>
```

```
<?php endif ?>

<?php if(empty($_SESSION['user_id'])) : ?>

    <?php for($i = 7; $i < 9; $i++) : ?>

        <li class="<?= $position === $links[$i]->LinkPath ? 'active' : '' ?>">

            <a href="<?= $links[$i]->LinkPath ?>">

                <?= $links[$i]->LinkName ?>

            </a>

        </li>

    <?php endfor ?>

<?php endif ?>

</ul>
```

post-comment.php

```
<?php

    $all_comments = $database->selectComments($_GET['id']);

    $parent_comments = [];

    foreach($all_comments as $comment)

    {

        if($comment->ParentId == null) array_push($parent_comments, $comment);

    }

}
```

```
?>

<?php foreach($parent_comments as $parent_comment) : ?>

    <div class="row parent-comment mb-3">

        <?php if($_SESSION['role'] === 1) : ?>

            <a class="delete-comment" href="/delete-comment?comment_id=?= $parent_comment->CommentId?>">

                <i class="material-icons red-text">close</i>

            </a>

        <?php endif ?>

        <div class="row">

            <div class="col">

        </div>

        <div class="col l6">

            <a href="/profile?username=?= $parent_comment->UserName ?>">

                <?= $parent_comment->FullName ?>

            </a>

            <br>

            <small>

                <?= $parent_comment->CreatedAt ?>

            </small>

        </div>

    </div>

endforeach
```

```
</div>

</div>

<div class="row mb-3">

    <div class="col l11 offset-l1">

        <p>

            <?= $parent_comment->CommentBody ?>

        </p>

    </div>

</div>

<?php if(isset($_SESSION['user_id'])) : ?>

    <div class="row">

        <div class="col l11 offset-l1">

            <div class="col">

                <p class="reply-button">Reply</p>

            </div>

            <?php if($_SESSION['user_id'] == $parent_comment->UserId ) : ?>

                <div class="col">

                    <a href="/delete-comment?comment_id=<?= $parent_comment->CommentId ?>">Delete</a>

                </div>

            <?php endif ?>

        </div>

    </div>

</div>
```

```

    </div>

    <div class="row scale-transition hide">

        <div class="col l8 offset-l4">

            <form action="/comment" method="POST">

                <div class="input-field">

                    <input type="text" name="comment_body" id="replyComment" />

                    <label for="replyComment">Reply to <?= $parent_comment->FullName ?></label>

                </div>

                <input type="hidden" name="user_id" value="<?= $_SESSION['user_id'] ?>" />

                <input type="hidden" name="post_id" value="<?= $parent_comment->PostId ?>" />

                <input type="hidden" name="parent_id" value="<?= $parent_comment->CommentId ?>" />

                <button type="submit" class="btn btn-small blue lighten-1 waves-effect waves-light">Reply</button>

            </form>

        </div>

    </div>

<?php endif ?>

<?php

    $reply_comments = $database->selectReplies($parent_comment->PostId, $parent_comment->CommentId);

    foreach($reply_comments as $reply_comment) :

        ?>

        <div class="row">

```



```

<div class="col s10 offset-s2 m10 offset-m2 l10 offset-l2 reply-comment">

    <div class="row">

        <div class="col l2">

        </div>

        <div class="col l6">

            <a href="/profile?username=<?= $reply_comment->UserName ?>">

                <?= $reply_comment->FullName ?>

            </a>

            <br>

            <small>

                <?= $reply_comment->CreatedAt ?>

            </small>

        </div>

        <?php if($_SESSION['role'] === 1) : ?>

            <a class="delete-reply" href="/delete-comment?comment_id=<?= $reply_comment->CommentId?>">

                <i class="material-icons red-text">close</i>

            </a>

        <?php endif ?>

    </div>

</div class="row">

```

```
<div class="col m10 offset-m2 l10 offset-l2">

    <p>

        <?= $reply_comment->CommentBody ?>

    </p>

</div>

</div>

<div class="row">

    <div class="col l8 offset-l2">

        <?php if($_SESSION['user_id'] == $parent_comment->UserId || $_SESSION['user_id'] ==
$reply_comment->UserId) : ?>

            <div class="col">

                <a href="/delete-comment?comment_id=<?= $reply_comment->CommentId ?>">Delete</a>

            </div>

            <?php endif ?>

        </div>

    </div>

</div>

</div>

<?php endforeach ?>

</div>

<?php endforeach ?>
```

```
<div class="comment-section center">

    <?php if(empty($comments)) : ?>

        <h5 class="mb-3">No comments found! <?= isset($_SESSION['user_id']) ? 'Be the first to comment.' : '' ?> </h5>

    <?php endif ?>

    <?php if(isset($_SESSION['user_id'])) : ?>

        <div class="row center">

            <form action="/comment" method="POST">

                <div class="input-field">

                    <textarea class="materialize-textarea" name="comment_body" id="commentBody" ></textarea>

                    <label for="commentBody">What do you think?</label>

                </div>

                <input type="hidden" name="user_id" value="<?= $_SESSION['user_id'] ?>" />

                <input type="hidden" name="post_id" value="<?= $_GET['id'] ?>" />

                <button type="submit" class="btn btn-small blue lighten-1 waves-effect waves-light">Comment</button>

            </form>

        </div>

    <?php else : ?>

        <h6 class="mb-3">Only <a href="/register">registered</a> users can comment!</h6>

    <?php endif ?>

</div>
```

controllers

author.php

```
<?php

$page_title = 'Author';

require 'resources/views/pages/author.view.php';
```

index.php

```
<?php

$page_title = 'Home';

require 'resources/views/pages/index.view.php';
```

surveys.php

```
<?php

$access_method = Request::getMethod();

if($access_method !== 'POST') Utilities::redirect('/index');

else

{

    $poll_id = $_POST['poll_id'];

    $answer = $_POST['poll_answer'];

    $user_id = $_POST['user_id'];
```

```
$user = $database->selectFilteredRecord('Users', 'UserId', $user_id);

$database->updateAnswer($answer, $poll_id, $user_id);

Utilities::redirect('/profile?username=' . $user->Username);

}
```

contact.php

```
<?php
```

```
$page_title = 'Contact';

$access_method = Request::getMethod();

if($access_method === 'GET') require 'resources/views/pages/contact.view.php';

else

{

    $sender_mail = $_POST['contact_mail'];

    $message_title = $_POST['message_title'];

    $message_body = $_POST['message_body'];

    $errors = [];

    if(!Utilities::checkRegex($sender_mail, 'Mail')) $errors['credentials'] = 'E-mail is not as expected!';
```

```
if(empty($message_body)) $errors['message'] = 'Message can\'t be empty!';

if(count($errors))

{

    require 'resources/views/pages/contact.view.php';

    end();

}

else

{

    $database->insertRecords('ContactMessages', [

        'SenderMail' => $sender_mail,

        'MessageTitle' => empty($message_title) ? 'No title' : $message_title,

        'MessageBody' => $message_body

    ]);

    Utilities::redirect('/contact?success=Your message was sent successfully!');

}

}
```

controllers/validations

verify.php

```
<?php

$page_title = 'Verification';

$token = $_GET['token'];

if(isset($token) )
{

    $user = $database->selectFilteredRecord('Users', 'UserToken', $token);

    if(isset($user))
    {

        $database->updateRecord('Users', 'UserVerified', 1, 'UserId', $user->UserId);

        require 'resources/views/authentication/verify.view.php';

    }

}

if($_SERVER['HTTP_REFERER'] == null) Utilities::redirect('/index');

else require 'resources/views/authentication/verify.view.php';
```

controllers/user

profile.php

```
<?php

$page_title = 'Profile';

if(empty($_GET['username'])) Utilities::redirect('/index');

else

{

    $user = $database->selectFilteredRecord('Users', 'UserName', $_GET['username']);

    if(empty($user)) Utilities::redirect('/');

    $role_name = $database->selectFilteredRecord('Roles', 'RoleId', $user->RoleId);

    $posts = $database->selectFilteredRecords('Posts', 'UserId', $user->UserId);

    require 'resources/views/user/profile.view.php';

}
```


survey.php

```
<?php

$access_method = Request::getMethod();

$page_title = 'Surveys';

if($access_method === 'GET' && isset($_GET['user_id']))

{

    session_start();

    $current_user = $_SESSION['user_id'];

    $remote_user = $_GET['user_id'];

    if($current_user != $remote_user) Utilities::redirect('/index');

    else

    {

        $polls = $database->selectPolls($current_user);

        require 'resources/views/user/survey.view.php';

    }

}

else Utilities::redirect('/index');
```

update-image.php

```
<?php

session_start();

$image_path = $_FILES['image_path']['name'];

$target = 'public/img/users/' . basename($image_path);

$cover = $database->selectFilteredRecord('Users', 'UserId', $_SESSION['user_id']);

$old_cover = $cover->ProfilePicture;

$database->updateRecord('Users', 'ProfilePicture', $image_path, 'UserId', $_SESSION['user_id']);

move_uploaded_file($_FILES['image_path']['tmp_name'], $target);

if($old_cover !== 'blank-profile.png') unlink("public/img/users/" . $cover['ProfilePicture']);

Utilities::redirect('/profile?username=' . $_SESSION['username']);
```

controllers/authentication

login.php

```
<?php

$page_title = 'Log In';

$access_method = Request::getMethod();

if($access_method === 'GET') require 'resources/views/authentication/login.view.php';

else

{

    $user_id = $_POST['user_id'];

    $user_password = $_POST['user_password'];

    $values = [];

    $errors = [];

    if(!Utilities::checkRegex($user_id, 'UserId')) $errors['credentials'] = 'Your ID is not as expected!';

    $values['credentials'] = $user_id;

    if(!Utilities::checkRegex($user_password, 'Password')) $errors['password'] = 'Password is not as expected!';
```

```
$canBeLogged = false;

$user = $database->selectUser($user_id);

if(empty($user)) $errors['credentials'] = 'User with given ID is not found. Try Again!';

else

{

    $errors = [];

    if(!$user->UserVerified) $errors['credentials'] = 'User with given ID, hasn\'t verified his identity yet!';

    else

    {

        $errors = [];

        if(!password_verify($user_password, $user->UserPassword)) $errors['password'] = 'Incorrect password! Try
again!';

        else

        {

            $errors = [];

            $canBeLogged = true;

        }

    }

}
```

```
if(count($errors))

{

    require 'resources/views/authentication/login.view.php';

    exit();

}

if($canBeLogged)

{

    session_start();

    $_SESSION['user_id'] = $user->UserId;

    $_SESSION['username'] = $user->Username;

    $_SESSION['role'] = $user->RoleId;

    Utilities::redirect('/index');

}

}
```

logout.php

```
<?php

session_start();

session_unset();

session_destroy();

Utilities::redirect('/index');

exit();
```

registration.php

```
<?php

$page_title = 'Register';

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\SMTP;

use PHPMailer\PHPMailer\Exception;

require 'vendor/autoload.php';

$accessMethod = Request::getMethod();
```

```
if($accessMethod === 'GET') require 'resources/views/authentication/registration.view.php';

else

{

    $first_name = $_POST['first_name'];

    $last_name = $_POST['last_name'];

    $username = $_POST['user_name'];

    $user_mail = $_POST['user_mail'];

    $user_password = $_POST['password'];

    $password_repeat = $_POST['re_password'];

    $users = $database->selectRecords('Users');

    $values = [];

    $errors = [];

    if(!Utilities::checkRegex($first_name, 'FirstName')) $errors['firstname'] = 'First name is not as expected!';

    $values['firstname'] = $first_name;

    if(!Utilities::checkRegex($last_name, 'LastName')) $errors['lastname'] = 'Last name is not as expected!';

    $values['lastname'] = $last_name;

    if(!Utilities::checkRegex($username, 'Username')) $errors['username'] = 'Username is not as expected!';

    $values['username'] = $username;
```

```
if(!Utilities::checkRegex($user_mail, 'Mail')) $errors['usermail'] = 'E-Mail is not as expected!';

$values['usermail'] = $user_mail;

if(!Utilities::checkRegex($user_password, 'Password')) $errors['password'] = 'Password is not as expected!';

if($user_password !== $password_repeat) $errors['re_password'] = 'Passwords don\'t match';


$verification_token = md5(time() . $username);


$found = false;

foreach($users as $user)

{

    if($user->Username === $username)

    {

        $errors['username'] = 'Username is already taken!';

        $found = true;

    }

    if($user->UserMail === $user_mail)

    {

        $errors['usermail'] = 'E-mail is already taken!';

        $found = true;

    }

}
```



```
if(count($errors))

{

    require 'resources/views/authentication/registration.view.php';

    exit();

}

if(!$found)

{

    $database->insertRecords('Users', [

        'FirstName' => $first_name,

        'LastName' => $last_name,

        'Username' => $username,

        'UserMail' => $user_mail,

        'UserPassword' => password_hash($user_password, PASSWORD_DEFAULT),

        'UserToken' => $verification_token

    ]);

    $mail = new PHPMailer(true);

    try

    {

        $mail->SMTPDebug = SMTP::DEBUG_SERVER;
```

```
$mail->isSMTP();

$mail->Host = 'smtp.gmail.com';

$mail->SMTPAuth = true;

$mail->Username = 'xxxxx';

$mail->Password = 'xxxxx';

$mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;

$mail->Port = 465;

$mail->setFrom('xxxxx');

$mail->addAddress($user_mail);

$mail->isHTML(true);

$mail->Subject = 'WireFrame - Account Verification';

$mail->Body = "Verify: <a href=\"http://localhost:8800/verify?token=$verification_token\">Verify</a>";

$mail->AltBody = "Click to verify: http://localhost:8800/verify?token=$verification_token";

$mail->send();

Utilities::Redirect('/verify');

}

catch (Exception $exception)
```

```
{  
  
    Utilities::dieDump($exception->getMessage());  
  
}  
  
}  
  
}
```

controllers/posts

comment.php

```
<?php

$comment = $_POST['comment_body'];

$user_id = $_POST['user_id'];

$post_id = $_POST['post_id'];

$parent_id = $_POST['parent_id'];

if(empty($comment)) Utilities::redirect('/view-post?id=' . $post_id . '&errors="Comment body can\'t be empty!');

else

{

    $database->insertRecords('Comments', [

        'ParentId' => $parent_id,

        'CommentBody' => htmlspecialchars($comment),

        'PostId' => $post_id,

        'UserId' => $user_id

    ]);

    Utilities::redirect('/view-post?id=' . $post_id);

}
```

create-post.php

```
<?php

$page_title = 'Create post';

$access_method = Request::getMethod();

$categories = $database->selectRecords('Categories');

if($access_method !== 'POST') require 'resources/views/posts/create-post.view.php';

else

{

    $post_title = $_POST['post_title'];

    $post_body = $_POST['post_body'];

    $image_path = $_FILES['image_path']['name'];

    $image_name = uniqid() . '_' . time();

    $target = 'public/img/covers/' . $image_name;

    $post_category = [];

    $values = [];

    $errors = [];

    if(empty($_POST['category_select'])) $errors['category'] = 'Please, provide at least one category for post!';

    else
```

```
{

    foreach($_POST['category_select'] as $selected)

    {

        array_push($post_category, $selected);

    }

}

if(empty($post_title))

{

    $errors['title'] = 'Please, fill in the post title!';

    if(!preg_match('/^[A-Za-z0-9-\?\s\#]{3,50}$/', $post_title)) $errors['title'] = 'Title is not as expected!';

}

else $values['title'] = $post_title;

if(empty($post_body)) $errors['body'] = 'Please, fill in the post body!';

else $values['body'] = $post_body;

if(empty($image_path)) $errors['image'] = 'Please, include a proper cover image!';

if(count($errors))

{

    require 'resources/views/posts/create-post.view.php';

}
```

```
        exit();

    }

    else

    {

        session_start();

        $database->insertRecords('Posts', [

            'PostTitle' => $post_title,

            'PostBody' => $post_body,

            'UserId' => $_SESSION['user_id']

        ]);

        $last_post = $database->selectLastRecords('Posts', 'PostId', 1);

        if(empty($last_post)) $post_id = 1;

        else $post_id = $last_post[0]->PostId;

        $database->insertRecords('CoverImages', [

            'CoverPath' => $image_name,

            'PostId' => $post_id

        ]);
```

```
move_uploaded_file($_FILES['image_path']['tmp_name'], $target);

foreach($post_category as $category)

{

    $database->insertRecords('PostCategory', [

        'PostId' => $post_id,

        'CategoryId' => (int) $category

    ]);

}

Utilities::redirect('/profile?username=' . $_SESSION['username']);

}

}
```

delete-comment.php

```
<?php

$comment_id = $_GET['comment_id'];

$writer = $database->selectFilteredRecord('Comments', 'CommentId', $comment_id);

session_start();
```



```
if(empty($comment_id)) Utilities::redirect('/index');

else if($_SESSION['role'] === 1 || $writer->UserId === $_SESSION['user_id'])

{

    $comment = $database->selectFilteredRecord('Comments', 'CommentId', $comment_id);

    $post_id = $comment->PostId;

    if($comment->ParentId === null)

    {

        $database->deleteRecords('Comments', 'ParentId', $comment_id);

        $database->deleteRecords('Comments', 'CommentId', $comment_id);

    }

    else $database->deleteRecords('Comments', 'CommentId', $comment_id);

    Utilities::redirect('/view-post?id=' . $post_id);

}
```

delete-post.php

```
<?php

$page_title = 'My profile';

$access_method = Request::getMethod();
```

```
if($access_method !== 'GET') Utilities::Redirect('/index');

else

{

    session_start();

    $id = $_GET['id'];

    $writer = $database->selectFilteredRecord('Posts', 'PostId', $id);

    $writer_id = $writer->UserId;

    if($writer_id === $_SESSION['user_id'] || $_SESSION['role'] === 1)

    {

        $cover = $database->selectFilteredRecord('CoverImages', 'PostId', $id);

        $database->deleteRecords('PostCategory', 'PostId', $id);

        $database->deleteRecords('CoverImages', 'PostId', $id);

        $database->deleteRecords('Comments', 'PostId', $id);

        $database->deleteRecords('Posts', 'PostId', $id);

        unlink("public/img/covers/" . $cover->CoverPath);

        session_start();

        Utilities::redirect('/profile?username=' . $_SESSION['username']);

    }

    else Utilities::Redirect('/index');
```

```
}
```

edit-post.php

```
<?php

$page_title = 'Edit post';

$access_method = Request::getMethod();

$categories = $database->selectRecords('Categories');

if($access_method === 'POST')
{

    $post_title = $_POST['post_title'];

    $post_body = $_POST['post_body'];

    $post_id = $_POST['post_id'];

    $old_path = $_POST['old_path'];

    $new_path = $_FILES['new_path']['name'];

    $image_name = uniqid() . '_' . time();

    $target = 'public/img/covers/' . $image_name;

    $post_category = [];
```

```
$values = [];  
  
$errors = [];  
  
if(empty($_POST['category_select'])) $errors['category'] = 'Please, provide at least one category for post!';  
  
else  
  
{  
  
    foreach($_POST['category_select'] as $selected)  
  
    {  
  
        array_push($post_category, $selected);  
  
    }  
  
}  
  
  
  
if(empty($post_title))  
  
{  
  
    $errors['title'] = 'Please, fill in the post title!';  
  
    if(!preg_match('/^[A-Za-z0-9\-\?\s\#]{3,50}$/ ', $post_title)) $errors['title'] = 'Title is not as expected!';  
  
}  
  
else $values['title'] = $post_title;  
  
  
  
if(empty($post_body)) $errors['body'] = 'Please, fill in the post body!';  
  
else $values['body'] = $post_body;
```

```
if(count($errors))

{

    require 'resources/views/posts/edit-post.view.php';

    exit();

}

else

{

    session_start();

    $database->updateRecord('Posts', 'PostTitle', $post_title, 'PostId', $post_id);

    $database->updatePostBody($post_body, $post_id);

    if($new_path !== '') {

        $database->updateRecord('CoverImages', 'CoverPath', $image_name, 'PostId', $post_id);

        move_uploaded_file($_FILES['new_path']['tmp_name'], $target);

        unlink('public/img/covers/' . $old_path);

    }

    $database->deleteRecords('PostCategory', 'PostId', $post_id);

    foreach($post_category as $category)

    {
```

```
        $database->insertRecords('PostCategory', [

            'PostId' => $post_id,

            'CategoryId' => $category

        ]);

    }

    Utilities::redirect('/view-post?id=' . $post_id);

}

}

else

{

    $post = $database->selectFilteredRecord('Posts', 'PostId', $_GET['id']);

    $old_categories = $database->selectFilteredRecords('PostCategory', 'PostId', $_GET['id']);

    $image = $database->selectFilteredRecord('CoverImages', 'PostId', $_GET['id']);

    $post_categories = [];

    foreach($old_categories as $category)

    {

        array_push($post_categories, $category->CategoryId);

    }

}
```

```
$values['title'] = $post->PostTitle;

$values['body'] = $post->PostBody;

$values['path'] = $image->CoverPath;


require 'resources/views/posts/edit-post.view.php';

}
```

posts.php

```
<?php

$page_title = 'Posts';

$access_method = Request::getMethod();

$categories = $database->selectRecords('Categories');

$recent_posts = $database->selectLastRecords('Posts', 'CreatedAt', 5, 1);

if($access_method === 'GET')

{

    $chosen_categories = [];

    $number_of_records = $database->selectRecords('Posts');

    $pages_number = ceil(count($number_of_records) / 5);
```

```
if(!isset($_GET['page'])) $start_point = 0;

else $start_point = ($_GET['page']-1) * 5;


if(isset($_GET['categories']))

{

    foreach($_GET['categories'] as $category_id) array_push($chosen_categories, $category_id);

    $posts = $database->selectPostsByCategory($chosen_categories, 0, 1, $start_point);

}

else $posts = $database->selectRecords('Posts', 1, 1, $start_point, 1);

require 'resources/views/pages/posts.view.php';


if($_GET['page'] > $pages_number) Utilities::redirect('/index');

}

else Utilities::redirect('/posts');
```


search.php

```
<?php

$search_query = $_GET['query'];

$suggested_posts = [];

$error = 'No posts found!';

$posts = $database->selectRecords('Posts');

if($search_query !== '')
{
    $search_query = strtolower($search_query);

    $query_length = strlen($search_query);

    foreach($posts as $post)
    {
        if(stristr(strtolower($post->PostTitle), $search_query)) array_push($suggested_posts, $post);
    }
}
```

```
echo !empty($suggested_posts) ? json_encode($suggested_posts) : $posts;
```

view-post.php

```
<?php

$post_id = $_GET['id'];

$post = $database->selectFilteredRecord('Posts', 'PostId', $post_id);

$cover = $database->selectFilteredRecord('CoverImages', 'PostId', $post_id);

$writer = $database->selectFilteredRecord('Users', 'UserId', $post->UserId);

$categories = $database->selectCategoriesOfPost($post_id);

$comments = $database->selectComments($_GET['id']);

$page_title = $post->PostTitle;

require 'resources/views/posts/view-post.view.php';
```

CSS

style.css

```
* {  
  
  margin: 0;  
  
  padding: 0;  
  
  -webkit-box-sizing: border-box;  
      box-sizing: border-box;  
  
}  
  
*:not(i.material-icons) {  
  
  font-family: 'Poppins', sans-serif;  
  
}  
  
* i, * a {  
  
  cursor: pointer;  
  
}  
  
html {
```

```
overflow-x: hidden;

}

html .m-a {

    margin: 0 auto !important;

}

html .mb-1 {

    margin-bottom: 1rem;

}

html .mb-2 {

    margin-bottom: 2rem;

}

html .mb-3 {

    margin-bottom: 3rem;

}

html .mb-5 {

    margin-bottom: 5rem !important;

}
```

```
html .p-0 {  
  
  padding: 0 !important;  
  
}
```

```
html .p-1 {  
  
  padding: 1rem !important;  
  
}
```

```
html .p-3 {  
  
  padding: 3rem !important;  
  
}
```

```
html .p-5 {  
  
  padding: 5rem !important;  
  
}
```

```
html .pt-1 {  
  
  padding-top: 1rem !important;  
  
}
```

```
html .pt-2 {
```

```
padding-top: 2rem !important;
}
```

```
html .pt-5 {

padding-top: 5rem !important;

}
```

```
html .pl-0 {

padding-left: 0 !important;

}
```

```
html .pl-1 {

padding-left: 1rem !important;

}
```

```
html .pl-2 {

padding-left: 2rem !important;

}
```

```
html .pr-0 {

padding-right: 0 !important;

}
```

```
html .mr-1 {  
  
  margin-right: 1rem !important;  
  
}  
  
html .blue.lighten-7 {  
  
  background-color: white !important;  
  
}  
  
html blockquote {  
  
  text-transform: uppercase;  
  
  border-left-color: #42a5f5 !important;  
  
}  
  
html body {  
  
  overflow-x: hidden;  
  
  position: relative;  
  
}  
  
html body .ms {  
  
  margin-bottom: 25rem !important;  
  
}
```

```
html body .user-profile {  
  
  width: 50px;  
  
  height: 50px;  
  
  border-radius: 50%;  
  
}
```

```
html body textarea#messageBody {  
  
  height: 300px;  
  
}
```

```
html body textarea:focus {  
  
  border-bottom-color: #42a5f5 !important;  
  
  -webkit-box-shadow: 0 1px 0 0 #42a5f5 !important;  
  
  box-shadow: 0 1px 0 0 #42a5f5 !important;  
  
}
```

```
html body textarea:focus + label {  
  
  color: #42a5f5 !important;  
  
}
```

```
html body .input-field .prefix.active {
```



```
color: #42a5f5;

}

html body .reply-button {

  cursor: pointer;

}

html body .parent-comment {

  position: relative;

  -webkit-box-shadow: 0 0 2px 0 grey;

  box-shadow: 0 0 2px 0 grey;

  padding: 1rem .5rem;

}

html body .parent-comment .row .reply-comment {

  position: relative;

  padding: 1rem;

  -webkit-box-shadow: 0 0 2px 0 grey;

  box-shadow: 0 0 2px 0 grey;

}

html body .parent-comment .row .reply-comment .delete-reply {
```

```
position: absolute;

top: 20px;

right: 20px;

}

html body .parent-comment .delete-comment {

position: absolute;

right: 10px;

top: 10px;

}

html body .prof-pic {

width: 60px;

height: 60px;

border-radius: 50%;

}

html body .page-name {

text-transform: capitalize;

}

html body .info-cards {
```

```
margin-bottom: 3rem;

}

html body .info-cards .stats-card {

  display: -webkit-box;

  display: -ms-flexbox;

  display: flex;

  -webkit-box-orient: vertical;

  -webkit-box-direction: normal;

  -ms-flex-direction: column;

  flex-direction: column;

  -webkit-box-align: center;

  -ms-flex-align: center;

  align-items: center;

  -webkit-box-pack: justify;

  -ms-flex-pack: justify;

  justify-content: space-between;

  padding: 2rem !important;

  margin-bottom: 3rem;

  border-radius: 10px;

  -webkit-transition: all 300ms ease !important;

  transition: all 300ms ease !important;
```

```
-webkit-box-shadow: 0 0 3px 0 #42a5f5 !important;

    box-shadow: 0 0 3px 0 #42a5f5 !important;
}

html body .info-cards .stats-card i {

    margin-bottom: 1rem;
}

html body .info-cards .stats-card p {

    margin-bottom: 1rem;

    text-transform: uppercase;
}

html body .info-cards .stats-card p:first-of-type {

    font-size: 2rem;
}

html body .info-cards .stats-card:hover {

    cursor: help;

    -webkit-box-shadow: 0 0 10px 0 #42a5f5 !important;

    box-shadow: 0 0 10px 0 #42a5f5 !important;
}
```

```
html body input.pass {  
  
    border-bottom-color: #42a5f5 !important;  
  
}  
  
html body input.pass + label {  
  
    color: #42a5f5 !important;  
  
}  
  
html body input[type="radio"]:checked + span::after {  
  
    border: 2px solid #42a5f5;  
  
    background-color: #42a5f5 !important;  
  
}  
  
html body input.fail {  
  
    border-bottom-color: #ff0000 !important;  
  
}  
  
html body input.fail + label {  
  
    color: #ff0000 !important;  
  
}
```

```
html body .verify-container {  
  
  margin: 150px auto;  
  
  padding: 3em;  
  
  border: 2px solid #42a5f5;  
  
  -webkit-box-shadow: 0 0 7px 0 #000;  
  
    box-shadow: 0 0 7px 0 #000;  
  
  display: -ms-grid;  
  
  display: grid;  
  
  place-content: center;  
  
  justify-items: center;  
  
}  
  
html body .verify-container > img {  
  
  width: 20%;  
  
}  
  
html body .verify-container h3 {  
  
  font-size: 1.5rem;  
  
}  
  
html body .verify-container h4 {  
  
  font-size: 1rem;
```

```
}
```

```
html body #userInfo #profilePictureContainer {
```

```
margin-bottom: 5rem;
```

```
text-align: center;
```

```
}
```

```
html body #userInfo #profilePictureContainer #profileImage {
```

```
margin-bottom: 1rem;
```

```
border-radius: 50%;
```

```
width: 200px;
```

```
height: 200px;
```

```
overflow: hidden;
```

```
}
```

```
html body #userInfo #profilePictureContainer #profileImage img {
```

```
width: 100%;
```

```
cursor: pointer;
```

```
}
```

```
html body #userInfo #userBio {
```

```
background-color: #eef8ff;
```

```
padding: 3rem 1.5rem;

border-radius: 10px;

}

html body .post-list {

padding: 1.5rem 2rem;

}

html body .recent-post {

margin-bottom: 1rem;

}

html body .recent-post p {

color: grey;

font-size: .85rem;

}

html body input[type="checkbox"]:checked + span:not(.lever):before {

border-right: 2px solid #42a5f5;

border-bottom: 2px solid #42a5f5;

}
```



```
html body #textarea1 {  
  
    height: 300px;  
  
}  
  
html body #textarea1:focus {  
  
    border-bottom-color: #42a5f5 !important;  
  
    -webkit-box-shadow: 0 1px 0 0 #42a5f5 !important;  
  
    box-shadow: 0 1px 0 0 #42a5f5 !important;  
  
}  
  
html body #textarea1:focus + label {  
  
    color: #42a5f5 !important;  
  
}  
  
html body input::-webkit-input-placeholder {  
  
    font-size: .75rem;  
  
}  
  
html body input:-ms-input-placeholder {  
  
    font-size: .75rem;  
  
}
```

```
html body input::-ms-input-placeholder {

    font-size: .75rem;

}

html body input::placeholder {

    font-size: .75rem;

}

html body input:focus {

    border-bottom: 1px solid #42a5f5 !important;

    -webkit-box-shadow: 0 1px 0 0 #42a5f5 !important;

    box-shadow: 0 1px 0 0 #42a5f5 !important;

}

html body input:focus + label {

    color: #42a5f5 !important;

}

html body .preloader {

    z-index: 1000000000;

    position: absolute;

    display: -webkit-box;
```

```
display: -ms-flexbox;

display: flex;

-webkit-box-align: center;

-ms-flex-align: center;

align-items: center;

-webkit-box-pack: center;

-ms-flex-pack: center;

justify-content: center;

top: 0;

left: 0;

right: 0;

bottom: 0;

}

html body .preloader .preloader-wrapper {

position: absolute;

top: 50vh;

}

html body .parallax-container {

height: 70vh;

display: -webkit-box;
```

```
display: -ms-flexbox;

display: flex;

-webkit-box-align: center;

-ms-flex-align: center;

align-items: center;

-webkit-box-pack: center;

-ms-flex-pack: center;

justify-content: center;
}

html body .cross-section {

width: 100%;

height: 25vh;

-webkit-clip-path: polygon(0 0, 100% 0, 100% 15%, 0 100%);

clip-path: polygon(0 0, 100% 0, 100% 15%, 0 100%);

background-color: #f1f8ff;
}

html body .opposite {

-webkit-clip-path: polygon(100% 0%, 100% 100%, 0% 100%, 100% 15%);

clip-path: polygon(100% 0%, 100% 100%, 0% 100%, 100% 15%);
}
```

```
html body footer .blog-info li {  
  
  padding-left: .8rem;  
  
  margin-bottom: .5rem;  
  
}  
  
html body footer .blog-info li a {  
  
  color: #fff;  
  
  font-size: 1.1rem;  
  
}  
  
/*# sourceMappingURL=style.css.map */
```

JavaScript

main.js

```
let urlPath = window.location.pathname

if(urlPath === '/register') validateRegistration()

if(urlPath === '/login') validateLogin()

if(urlPath === '/reset') resetPasswordRequest()

if(urlPath === '/posts') searchAjax()

if(urlPath === '/dashboard/modify') categoryModify()

if(urlPath === '/dashboard/surveys') addQuestion()

// VALIDATE REGISTER FORM

function validateRegistration() {

    const firstNameRegex = /^[A-Z]{1}[a-z]{2,14}$/

    const lastNameRegex = /^[A-Z]{1}[a-z]{4,29}$/

    const passwordRegex = /^[A-Z]{1}[a-z0-9!@#$$%^.&*]{7,19}$/
```

```
const userMailRegex = /^[a-zA-Z0-9]([a-z]|[0-9])+\.?-?_?([a-z]|[0-9])*\.?([a-z]|[0-9])*@[a-z]{3,}\.([a-z]{2,4}\.)?([a-z]{2,4})$/

const userNameRegex = /^[a-z]{1}[a-z0-9]{4,29}$/

const registrationForm = document.getElementById('registrationForm')

const firstName = document.getElementById('firstName')

const lastName = document.getElementById('lastName')

const userPassword = document.getElementById('password')

const passwordRepeat = document.getElementById('rePassword')

const userMail = document.getElementById('userMail')

const userName = document.getElementById('userName')

registrationForm.addEventListener('submit', (event) => {

  if(!registerButton()) {

    event.preventDefault()

    checkField(firstName, 'First name', firstNameRegex)

    checkField(lastName, 'Last name', lastNameRegex)

    checkField(userPassword, 'Password', passwordRegex)

    checkField(passwordRepeat, 'Repeated password', passwordRegex)

    checkField(userMail, 'Mail', userMailRegex)

    checkField(userName, 'Username', userNameRegex)

    checkPassword(userPassword.value, passwordRepeat.value)
```

```
    }

  })

  function registerButton() {

    if(

      checkField(firstName, 'First name', firstNameRegex) &&

      checkField(lastName, 'Last name', lastNameRegex) &&

      checkField(userPassword, 'Password', passwordRegex) &&

      checkField(passwordRepeat, 'Repeated password', passwordRegex) &&

      checkField(userMail, 'Mail', userMailRegex) &&

      checkField(userName, 'Username', userNameRegex) &&

      checkPassword(userPassword.value, passwordRepeat.value)

    ) return 1

    return 0

  }

  function checkPassword(password, repeat) {

    let infoText = passwordRepeat.nextElementSibling.nextElementSibling

    if(password !== repeat || repeat === '') {

      infoText.className = 'red-text'

      infoText.innerHTML = 'Repeated password does not match!'

      return 0

    }
```



```
    }

    return 1

}

addListenerForField(firstName, 'First name', firstNameRegex, 'keyup')

addListenerForField(firstName, 'First name', firstNameRegex, 'blur')

addListenerForButton(firstName, registerButton, 'keyup')


addListenerForField.lastName, 'Last name', lastNameRegex, 'keyup')

addListenerForField.lastName, 'Last name', lastNameRegex, 'blur')

addListenerForButton.lastName, registerButton, 'keyup')


addListenerForField.userName, 'Username', userNameRegex, 'keyup')

addListenerForField.userName, 'Username', userNameRegex, 'blur')

addListenerForButton.userName, registerButton, 'keyup')


addListenerForField.userMail, 'Mail', userMailRegex, 'keyup')

addListenerForField.userMail, 'Mail', userMailRegex, 'blur')

addListenerForButton.userMail, registerButton, 'keyup')


addListenerForField.userPassword, 'Password', passwordRegex, 'keyup')

addListenerForField.userPassword, 'Password', passwordRegex, 'blur')
```

```
addListenerForButton(userPassword, registerButton, 'keyup')

passwordRepeat.addEventListener('keyup', checkPassword)

addListenerForField(passwordRepeat, 'Repeated password', passwordRegex, 'keyup')

addListenerForField(passwordRepeat, 'Repeated password', passwordRegex, 'blur')

addListenerForButton(passwordRepeat, registerButton, 'keyup')

}

// END OF VALIDATE REGISTER FORM

// VALIDATE LOGIN FORM

function validateLogin() {

    const userIdRegex = /^[a-zA-Z0-9]([a-z]|[0-9])+\.?[-?_]?([a-z]|[0-9])*\.?([a-z]|[0-9])*@[a-z]{3,}\.([a-z]{2,4}\.)?([a-z]{2,4}))|(([a-z]{1})[a-z0-9]{4,29}))$/

    const passwordRegex = /^[A-Z]{1}[a-z0-9!@#$$%^.&*]{7,19}$/

    const loginForm = document.getElementById('loginForm')

    const userId = document.getElementById('userId')

    const userPassword = document.getElementById('userPassword')

    addListenerForButton(userId, loginButton, 'keyup')
```

```
addListenerForField(userId, 'Your ID', userIdRegex, 'blur')

addListenerForField(userId, 'Your ID', userIdRegex, 'keyup')


addListenerForField(userPassword, 'Password', passwordRegex, 'keyup')

addListenerForField(userPassword, 'Password', passwordRegex, 'blur')

addListenerForButton(userPassword, loginButton, 'keyup')
```

```
function loginButton() {

    if(

        checkField(userId, 'Your ID', userIdRegex) &&

        checkField(userPassword, 'Password', passwordRegex)

    ) return 1

    return 0

}

}
```

```
// END OF VALIDATE LOGIN FORM
```

```
// VALIDATE PASSWORD REQUEST
```

```
function resetPasswordRequest() {
```

```
const userMailRegex = /^[a-zA-Z0-9]([a-z]|[0-9])+\.?-?_?([a-z]|[0-9])*\.?([a-z]|[0-9])*@[a-z]{3,}\.([a-z]{2,4}\.)?([a-z]{2,4})$/

const passwordResetForm = document.getElementById('resetForm')

const mailField = document.getElementById('userMail')

passwordResetForm.addEventListener('submit', (event) => {

  if(!resetBtn()) {

    checkField(mailField, 'E-mail', userMailRegex)

    event.preventDefault()

  }

})

function resetBtn() {

  if(checkField(mailField, 'E-mail', userMailRegex)) return 1

  return 0

}

addListenerForButton(mailField, resetBtn, 'keyup')

addListenerForField(mailField, 'E-mail', userMailRegex, 'blur')

addListenerForField(mailField, 'E-mail', userMailRegex, 'keyup')

}
```

```
// END OF VALIDATE PASSWORD REQUEST

// FORM VALIDATION METHODS

function fieldValid(field) {

    field.nextElementSibling.nextElementSibling.innerHTML = ''

}

function fieldInvalid(field, text) {

    field.nextElementSibling.nextElementSibling.innerText = `${text}`

    field.nextElementSibling.nextElementSibling.classList.add('red-text')

}

function checkField(field, fieldName, expression) {

    let fieldValue = field.value

    return checkRegEx(expression, fieldValue, fieldName, field)

}

function checkRegEx(expression, fieldValue, fieldName, field) {

    if(expression.test(String(fieldValue))) {

        fieldValid(field)

    }

}
```

```
        return 1 // Is valid, return 1

    } else {

        fieldInvalid(field, `${fieldName} is not as expected!`)

        return 0 // It isn't as expected, return 0

    }

}

function checkBtn(callback) {

    if(callback()) return 1

    return 0

}

function addListenerForField(field, fieldName, expression, event) {

    field.addEventListener(event, () => {

        checkField(field, fieldName, expression)

    })

}

function addListenerForButton(field, callback, event) {

    field.addEventListener(event, () => {

        checkBtn(callback)

    })

}
```

```
}

// END OF FORM VALIDATION METHODS

// ADD Categories

function categoryModify() {

    document.getElementById('addCategoryBtn').addEventListener('click', () => {

        let inputField = document.getElementById('addCategoryField')

        if(inputField.classList.contains('hide')) {

            inputField.classList.remove('hide')

        } else inputField.classList.add('hide')

    })

}

// END OF ADD Categories

// DEACTIVATE POLL

document.querySelectorAll('.deactivate-survey').forEach(button => {

    button.addEventListener('click', (event) => {

        let surveyId = event.target.parentElement.parentElement.children[0].innerText

        let data = {survey_id: surveyId}

        let xhr = new XMLHttpRequest()

        xhr.open('POST', '/dashboard/surveys', true)
```

```
xhr.setRequestHeader('Content-type', 'application/json')

xhr.send(JSON.stringify(data))

event.target.className = 'red-text material-icons'

event.target.innerText = 'close'

})

})

// END OF DEACTIVATE POLL


// APPROVE POST

document.querySelectorAll('.approve-post').forEach(button => {

  button.addEventListener('click', (event) => {

    let postId = event.target.parentElement.parentElement.children[0].innerText

    let data = {post_id: postId}

    let xhr = new XMLHttpRequest()

    xhr.open('POST', '/dashboard/posts', true)

    xhr.setRequestHeader('Content-type', 'application/json')

    xhr.send(JSON.stringify(data))

    event.target.className = 'green-text'

    event.target.innerText = 'Approved'

  })

})

// END OF APPROVE POST
```



```
// CHANGE ROLE

document.querySelectorAll('.role-change').forEach(list => {

  list.addEventListener('change', (event) => {

    let selectedRole = event.target.value

    let user = event.target.parentElement.parentElement.parentElement.children[0].innerText

    let data = {

      role_id: selectedRole,

      user_id: user

    }

    let xhr = new XMLHttpRequest()

    xhr.open('POST', '/dashboard/users', true)

    xhr.setRequestHeader('Content-type', 'application/json')

    xhr.send(JSON.stringify(data))

  })

})

// END OF CHANGE ROLE


// ADD QUESTION

function addQuestion() {

  let questionFields = document.getElementById('addQuestionField')
```

```
document.getElementById('addQuestion').addEventListener('click', () => {

    if(questionFields.classList.contains('hide')) {

        questionFields.classList.remove('hide')

    } else questionFields.classList.add('hide')

})

}

// END OF ADD QUESTION


// REPLY

document.querySelectorAll('.reply-button').forEach(btn => {

    btn.addEventListener('click', (event) => {

        let clicked = event.target

        let replyField = clicked.parentElement.parentElement.nextElementSibling

        if(replyField.classList.contains('hide')) {

            replyField.classList.remove('scale-out')

            replyField.classList.remove('hide')

            replyField.classList.add('scale-in')

        } else {

            replyField.classList.remove('scale-in')

            replyField.classList.add('scale-out')
```

```
        replyField.classList.add('hide')

    }

    })

    })

// END OF REPLY


// AJAX FOR WRITING DATA FROM DATABASE


function searchAjax() {

    document.getElementById('searchPosts').addEventListener('keyup', searchForPosts)

    const postsContainer = document.getElementById('postContainer')

    document.addEventListener('DOMContentLoaded', loadPosts)


    async function searchForPosts(event) {

        let query = event.target.value

        let postLink = ''

        let response = (await fetch('/search?query=' + query)).json()

        response.then(posts => {

            for(let post of posts) {

                postLink += `

                <tr>
```

```
        <td class="post-list">

            <a href="/view-post?id=${post.PostId}">

                ${post.PostTitle}

            </a>

        </td>

    </tr>

</tr>

</tr>

}

postsContainer.innerHTML = postLink ? postLink : 'No posts found'

}))

}
```

```
async function loadPosts() {

    let response = (await fetch('/posts')).json()

    let postLink = ''

    response.then(posts => {

        for(let post of posts) {

            postLink += `

                <tr>

                    <td class="post-list">

                        <a href="/view-post?id=${post.PostId}">

                            ${post.PostTitle}
```

```
                </a>

            </td>

        </tr>
    }

    postsContainer.innerHTML = postlink ? postlink : 'No posts found!'

})

}

}

// END OF AJAX FOR WRITING DATA FROM DATABASE


function pageLoader() {

    setTimeout(() => {

        $('.preloader').fadeOut();

    }, 2000)

    setTimeout(() => {

        $('.preloader').remove();

    }, 4000)

}
```

```
// MATERIALIZE & JQUERY DEPENDENCIES LOAD

$(document).ready(() => {

    pageLoader()

    $('select').formSelect()

    $('.materialboxed').materialbox()

    $('.sidenav').sidenav()

    $('.parallax').parallax()

    $('.modal').modal()

})

// END OF MATERIALIZE & JQUERY DEPENDENCIES LOAD
```

WireFrame