



INFORMATICS
INSTITUTE OF
TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY
In Collaboration with
UNIVERSITY OF WESTMINSTER

Coursework 2

Course
Software Engineering

Module
6COSC022C Advanced Server-Side Web Programming

Module Leader
Mr. Sriyan Fernando

By
Mr. Thaveesha Dheersekera

Date
January 19, 2023

1.

URL

<https://w1761369.users.ecs.westminster.ac.uk/developer-support/>

Code

config.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*
|-----
| Base Site URL
|-----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|   http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try to guess the protocol and
| path to your installation, but due to security concerns the hostname will
| be set to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|
*/

// localhost
// $config['base_url'] = 'http://localhost/developer-support/';

// UoW Server
$config['base_url'] = 'https://w1761369.users.ecs.westminster.ac.uk/developer-support/';

/*
|-----
| Index File
|-----
|
|
```

```

| Typically this will be your index.php file, unless you've renamed it to
| something else. If you are using mod_rewrite to remove the page set this
| variable so that it is blank.
|
*/
$config['index_page'] = 'index.php';

/*
|-----
| URI PROTOCOL
|-----
|
| This item determines which server global should be used to retrieve the
| URI string. The default setting of 'REQUEST_URI' works for most servers.
| If your links do not seem to work, try one of the other delicious flavors:
|
| 'REQUEST_URI'  Uses $_SERVER['REQUEST_URI']
| 'QUERY_STRING' Uses $_SERVER['QUERY_STRING']
| 'PATH_INFO'    Uses $_SERVER['PATH_INFO']
|
| WARNING: If you set this to 'PATH_INFO', URIs will always be URL-decoded!
*/
$config['uri_protocol'] = 'REQUEST_URI';

/*
|-----
| URL suffix
|-----
|
| This option allows you to add a suffix to all URLs generated by CodeIgniter.
| For more information please see the user guide:
|
| https://codeigniter.com/userguide3/general/urls.html
|
| Note: This option is ignored for CLI requests.
*/
$config['url_suffix'] = '';

/*
|-----
| Default Language
|-----
|
| This determines which set of language files should be used. Make sure
| there is an available translation if you intend to use something other
| than english.

```

```
|
*/
$config['language'] = 'english';

/*
|-----
| Default Character Set
|-----
|
| This determines which character set is used by default in various methods
| that require a character set to be provided.
|
| See http://php.net/htmlspecialchars for a list of supported charsets.
|
*/
$config['charset'] = 'UTF-8';

/*
|-----
| Enable/Disable System Hooks
|-----
|
| If you would like to use the 'hooks' feature you must enable it by
| setting this variable to TRUE (boolean). See the user guide for details.
|
*/
$config['enable_hooks'] = FALSE;

/*
|-----
| Class Extension Prefix
|-----
|
| This item allows you to set the filename/classname prefix when extending
| native libraries. For more information please see the user guide:
|
| https://codeigniter.com/userguide3/general/core_classes.html
| https://codeigniter.com/userguide3/general/creating_libraries.html
|
*/
$config['subclass_prefix'] = 'MY_';

/*
|-----
| Composer auto-loading
|-----
```

```

|
| Enabling this setting will tell CodeIgniter to look for a Composer
| package auto-loader script in application/vendor/autoload.php.
|
| $config['composer_autoload'] = TRUE;
|
| Or if you have your vendor/ directory located somewhere else, you
| can opt to set a specific path as well:
|
| $config['composer_autoload'] = '/path/to/vendor/autoload.php';
|
| For more information about Composer, please visit http://getcomposer.org/
|
| Note: This will NOT disable or override the CodeIgniter-specific
| autoloading (application/config/autoload.php)
*/
$config['composer_autoload'] = FALSE;

/*
|-----
| Allowed URL Characters
|-----
|
| This lets you specify which characters are permitted within your URLs.
| When someone tries to submit a URL with disallowed characters they will
| get a warning message.
|
| As a security measure you are STRONGLY encouraged to restrict URLs to
| as few characters as possible. By default only these are allowed: a-z 0-9~%.:_-
|
| Leave blank to allow all characters -- but only if you are insane.
|
| The configured value is actually a regular expression character group
| and it will be executed as: ! preg_match('/^[<permitted_uri_chars>]+$/',
|
| DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!
|
*/
$config['permitted_uri_chars'] = 'a-z 0-9~%.:_-';

/*
|-----
| Enable Query Strings
|-----
|
| By default CodeIgniter uses search-engine friendly segment based URLs:

```

```

| example.com/who/what/where/
|
| You can optionally enable standard query string based URLs:
| example.com?who=me&what=something&where=here
|
| Options are: TRUE or FALSE (boolean)
|
| The other items let you set the query string 'words' that will
| invoke your controllers and its functions:
| example.com/index.php?c=controller&m=function
|
| Please note that some of the helpers won't work as expected when
| this feature is enabled, since CodeIgniter is designed primarily to
| use segment based URLs.
|
*/
$config['enable_query_strings'] = FALSE;
$config['controller_trigger'] = 'c';
$config['function_trigger'] = 'm';
$config['directory_trigger'] = 'd';

/*
|-----
| Allow $_GET array
|-----
|
| By default CodeIgniter enables access to the $_GET array. If for some
| reason you would like to disable it, set 'allow_get_array' to FALSE.
|
| WARNING: This feature is DEPRECATED and currently available only
|         for backwards compatibility purposes!
|
*/
$config['allow_get_array'] = TRUE;

/*
|-----
| Error Logging Threshold
|-----
|
| You can enable error logging by setting a threshold over zero. The
| threshold determines what gets logged. Threshold options are:
|
| 0 = Disables logging, Error logging TURNED OFF
| 1 = Error Messages (including PHP errors)
| 2 = Debug Messages

```

```
| 3 = Informational Messages
| 4 = All Messages
|
| You can also pass an array with threshold levels to show individual error types
|
| array(2) = Debug Messages, without Error Messages
|
| For a live site you'll usually only enable Errors (1) to be logged otherwise
| your log files will fill up very fast.
|
*/
$config['log_threshold'] = 0;

/*
|-----
| Error Logging Directory Path
|-----
|
| Leave this BLANK unless you would like to set something other than the default
| application/logs/ directory. Use a full server path with trailing slash.
|
*/
$config['log_path'] = '';

/*
|-----
| Log File Extension
|-----
|
| The default filename extension for log files. The default 'php' allows for
| protecting the log files via basic scripting, when they are to be stored
| under a publicly accessible directory.
|
| Note: Leaving it blank will default to 'php'.
|
*/
$config['log_file_extension'] = '';

/*
|-----
| Log File Permissions
|-----
|
| The file system permissions to be applied on newly created log files.
|
| IMPORTANT: This MUST be an integer (no quotes) and you MUST use octal
```

```

|         integer notation (i.e. 0700, 0644, etc.)
*/
$config['log_file_permissions'] = 0644;

/*
|-----
| Date Format for Logs
|-----
|
| Each item that is logged has an associated date. You can use PHP date
| codes to set your own date formatting
|
*/
$config['log_date_format'] = 'Y-m-d H:i:s';

/*
|-----
| Error Views Directory Path
|-----
|
| Leave this BLANK unless you would like to set something other than the default
| application/views/errors/ directory. Use a full server path with trailing slash.
|
*/
$config['error_views_path'] = '';

/*
|-----
| Cache Directory Path
|-----
|
| Leave this BLANK unless you would like to set something other than the default
| application/cache/ directory. Use a full server path with trailing slash.
|
*/
$config['cache_path'] = '';

/*
|-----
| Cache Include Query String
|-----
|
| Whether to take the URL query string into consideration when generating
| output cache files. Valid options are:
|
| FALSE = Disabled

```



```
| TRUE      = Enabled, take all query parameters into account.  
|           Please be aware that this may result in numerous cache  
|           files generated for the same page over and over again.  
| array('q') = Enabled, but only take into account the specified list  
|           of query parameters.  
|  
|  
*/
```

```
$config['cache_query_string'] = FALSE;
```

```
/*  
|-----  
| Encryption Key  
|-----  
|  
| If you use the Encryption class, you must set an encryption key.  
| See the user guide for more info.  
|  
| https://codeigniter.com/userguide3/libraries/encryption.html  
|  
*/
```

```
$config['encryption_key'] = '';
```

```
/*  
|-----  
| Session Variables  
|-----  
|  
| 'sess_driver'  
|  
| The storage driver to use: files, database, redis, memcached  
|  
| 'sess_cookie_name'  
|  
| The session cookie name, must contain only [0-9a-z_-] characters  
|  
| 'sess_samesite'  
|  
| Session cookie SameSite attribute: Lax (default), Strict or None  
|  
| 'sess_expiration'  
|  
| The number of SECONDS you want the session to last.  
| Setting to 0 (zero) means expire when the browser is closed.  
|  
| 'sess_save_path'  
|
```

```

| The location to save sessions to, driver dependent.
|
| For the 'files' driver, it's a path to a writable directory.
| WARNING: Only absolute paths are supported!
|
| For the 'database' driver, it's a table name.
| Please read up the manual for the format with other session drivers.
|
| IMPORTANT: You are REQUIRED to set a valid save path!
|
| 'sess_match_ip'
|
| Whether to match the user's IP address when reading the session data.
|
| WARNING: If you're using the database driver, don't forget to update
|         your session table's PRIMARY KEY when changing this setting.
|
| 'sess_time_to_update'
|
| How many seconds between CI regenerating the session ID.
|
| 'sess_regenerate_destroy'
|
| Whether to destroy session data associated with the old session ID
| when auto-regenerating the session ID. When set to FALSE, the data
| will be later deleted by the garbage collector.
|
| Other session cookie settings are shared with the rest of the application,
| except for 'cookie_prefix' and 'cookie_httponly', which are ignored here.
|
| */
$config['sess_driver'] = 'files';
$config['sess_cookie_name'] = 'ci_session';
$config['sess_samesite'] = 'Lax';
$config['sess_expiration'] = 7200;
$config['sess_save_path'] = NULL;
$config['sess_match_ip'] = FALSE;
$config['sess_time_to_update'] = 300;
$config['sess_regenerate_destroy'] = FALSE;

/*
|-----
| Cookie Related Variables
|-----
|
| 'cookie_prefix' = Set a cookie name prefix if you need to avoid collisions

```

```

| 'cookie_domain' = Set to .your-domain.com for site-wide cookies
| 'cookie_path'   = Typically will be a forward slash
| 'cookie_secure' = Cookie will only be set if a secure HTTPS connection exists.
| 'cookie_httponly' = Cookie will only be accessible via HTTP(S) (no javascript)
| 'cookie_samesite' = Cookie's samesite attribute (Lax, Strict or None)
|
| Note: These settings (with the exception of 'cookie_prefix' and
|       'cookie_httponly') will also affect sessions.
|
*/
$config['cookie_prefix'] = "";
$config['cookie_domain'] = "";
$config['cookie_path']   = '/';
$config['cookie_secure'] = FALSE;
$config['cookie_httponly'] = FALSE;
$config['cookie_samesite'] = 'Lax';

/*
|-----
| Standardize newlines
|-----
|
| Determines whether to standardize newline characters in input data,
| meaning to replace \r\n, \r, \n occurrences with the PHP_EOL value.
|
| WARNING: This feature is DEPRECATED and currently available only
|         for backwards compatibility purposes!
|
*/
$config['standardize_newlines'] = FALSE;

/*
|-----
| Global XSS Filtering
|-----
|
| Determines whether the XSS filter is always active when GET, POST or
| COOKIE data is encountered
|
| WARNING: This feature is DEPRECATED and currently available only
|         for backwards compatibility purposes!
|
*/
$config['global_xss_filtering'] = FALSE;

/*

```

```

|-----
| Cross Site Request Forgery
|-----
| Enables a CSRF cookie token to be set. When set to TRUE, token will be
| checked on a submitted form. If you are accepting user data, it is strongly
| recommended CSRF protection be enabled.
|
| 'csrf_token_name' = The token name
| 'csrf_cookie_name' = The cookie name
| 'csrf_expire' = The number in seconds the token should expire.
| 'csrf_regenerate' = Regenerate token on every submission
| 'csrf_exclude_uris' = Array of URIs which ignore CSRF checks
*/
$config['csrf_protection'] = FALSE;
$config['csrf_token_name'] = 'csrf_test_name';
$config['csrf_cookie_name'] = 'csrf_cookie_name';
$config['csrf_expire'] = 7200;
$config['csrf_regenerate'] = TRUE;
$config['csrf_exclude_uris'] = array();

/*
|-----
| Output Compression
|-----
|
| Enables Gzip output compression for faster page loads. When enabled,
| the output class will test whether your server supports Gzip.
| Even if it does, however, not all browsers support compression
| so enable only if you are reasonably sure your visitors can handle it.
|
| Only used if zlib.output_compression is turned off in your php.ini.
| Please do not use it together with httpd-level output compression.
|
| VERY IMPORTANT: If you are getting a blank page when compression is enabled it
| means you are prematurely outputting something to your browser. It could
| even be a line of whitespace at the end of one of your scripts. For
| compression to work, nothing can be sent before the output buffer is called
| by the output class. Do not 'echo' any values with compression enabled.
|
*/
$config['compress_output'] = FALSE;

/*
|-----
| Master Time Reference
|-----

```

```

|
| Options are 'local' or any PHP supported timezone. This preference tells
| the system whether to use your server's local time as the master 'now'
| reference, or convert it to the configured one timezone. See the 'date
| helper' page of the user guide for information regarding date handling.
|
*/
$config['time_reference'] = 'local';

/*
|-----
| Rewrite PHP Short Tags
|-----
|
| If your PHP installation does not have short tag support enabled CI
| can rewrite the tags on-the-fly, enabling you to utilize that syntax
| in your view files. Options are TRUE or FALSE (boolean)
|
| Note: You need to have eval() enabled for this to work.
|
*/
$config['rewrite_short_tags'] = FALSE;

/*
|-----
| Reverse Proxy IPs
|-----
|
| If your server is behind a reverse proxy, you must whitelist the proxy
| IP addresses from which CodeIgniter should trust headers such as
| HTTP_X_FORWARDED_FOR and HTTP_CLIENT_IP in order to properly identify
| the visitor's IP address.
|
| You can use both an array or a comma-separated list of proxy addresses,
| as well as specifying whole subnets. Here are a few examples:
|
| Comma-separated: '10.0.1.200,192.168.5.0/24'
| Array:          array('10.0.1.200', '192.168.5.0/24')
|
*/
$config['proxy_ips'] = '';

```

database.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*

```

----- DATABASE CONNECTIVITY SETTINGS -----

This file will contain the settings needed to access your database.

For complete instructions please consult the 'Database Connection' page of the User Guide.

----- EXPLANATION OF VARIABLES -----

['dsn'] The full DSN string describe a connection to the database.

['hostname'] The hostname of your database server.

['username'] The username used to connect to the database

['password'] The password used to connect to the database

['database'] The name of the database you want to connect to

['dbdriver'] The database driver. e.g.: mysqli.

Currently supported:

cubrid, ibase, mssql, mysql, mysqli, oci8,

odbc, pdo, postgre, sqlite, sqlite3, sqlsrv

['dbprefix'] You can add an optional prefix, which will be added to the table name when using the Query Builder class

['pconnect'] TRUE/FALSE - Whether to use a persistent connection

['db_debug'] TRUE/FALSE - Whether database errors should be displayed.

['cache_on'] TRUE/FALSE - Enables/disables query caching

['cachedir'] The path to the folder where cache files should be stored

['char_set'] The character set used in communicating with the database

['dbcollat'] The character collation used in communicating with the database

NOTE: For MySQL and MySQLi databases, this setting is only used as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7 (and in table creation queries made with DB Forge).

There is an incompatibility in PHP with mysql_real_escape_string() which can make your site vulnerable to SQL injection if you are using a multi-byte character set and are running versions lower than these.

Sites using Latin-1 or UTF-8 database character set and collation are unaffected.

['swap_pre'] A default table prefix that should be swapped with the dbprefix

['encrypt'] Whether or not to use an encrypted connection.

'mysql' (deprecated), 'sqlsrv' and 'pdo/sqlsrv' drivers accept TRUE/FALSE

'mysqli' and 'pdo/mysqli' drivers accept an array with the following options:

'ssl_key' - Path to the private key file

'ssl_cert' - Path to the public key certificate file

'ssl_ca' - Path to the certificate authority file

'ssl_capath' - Path to a directory containing trusted CA certificates in PEM format

```

|         'ssl_cipher' - List of *allowed* ciphers to be used for the encryption, separated by colons (':')
|         'ssl_verify' - TRUE/FALSE; Whether verify the server certificate or not
|
|     ['compress'] Whether or not to use client compression (MySQL only)
|     ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
|                 - good for ensuring strict SQL while developing
|     ['ssl_options'] Used to set various SSL options that can be used when making SSL connections.
|     ['failover'] array - A array with 0 or more data for connections if the main should fail.
|     ['save_queries'] TRUE/FALSE - Whether to "save" all executed queries.
|
|         NOTE: Disabling this will also effectively disable both
|         $this->db->last_query() and profiling of DB queries.
|         When you run a query, with this setting set to TRUE (default),
|         CodeIgniter will store the SQL statement for debugging purposes.
|         However, this may cause high memory usage, especially if you run
|         a lot of SQL queries ... disable this to avoid that problem.
|
| The $active_group variable lets you choose which connection group to
| make active. By default there is only one group (the 'default' group).
|
| The $query_builder variable lets you determine whether or not to load
| the query builder class.
*/
$active_group = 'default';
$query_builder = TRUE;

// $db['default'] = array(
//     'dsn' => "",
//     'hostname' => 'localhost',
//     'port' => '3306',
//     'username' => 'root',
//     'password' => "",
//     'database' => 'w1761369_0',
//     'dbdriver' => 'mysqli',
//     'dbprefix' => "",
//     'pconnect' => FALSE,
//     'db_debug' => (ENVIRONMENT !== 'production'),
//     'cache_on' => FALSE,
//     'cachedir' => "",
//     'char_set' => 'utf8',
//     'dbcollat' => 'utf8_general_ci',
//     'swap_pre' => "",
//     'encrypt' => FALSE,
//     'compress' => FALSE,
//     'stricton' => FALSE,
//     'failover' => array(),
//     'save_queries' => TRUE

```

```
//);

// UoW database credentials

$db['default'] = array(
    'dsn' => "",
    'hostname' => 'phpmyadmin.ecs.westminster.ac.uk',
    'port' => '3306',
    'username' => 'w1761369',
    'password' => 'ChRswpdfTtV',
    'database' => 'w1761369_0',
    'dbdriver' => 'mysqli',
    'dbprefix' => "",
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => "",
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => "",
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

api/Questions.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

use Restserver\Libraries\REST_Controller;
require APPPATH . 'libraries/REST_Controller.php';

class Questions extends REST_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->database(); // load database
        $this->load->model('Questions_Model');
        $this->load->library('form_validation');
        $this->load->helper('security');
    }
}
```



```

public function index_get() {
    $questions = $this->Questions_Model->listQuestions();

    if($questions > 0) {
        $this->response(
            array(
                'status' => 1,
                'message' => 'Successfully listed all questions',
                'data' => $questions
            ), REST_Controller::HTTP_OK
        );
    } else {
        $this->response(
            array(
                'status' => 0,
                'message' => 'Failed to questions',
                'data' => $questions
            ), REST_Controller::HTTP_NOT_FOUND
        );
    }
}

public function question_get() {
    $id = $this->input->get('id');

    $question = $this->Questions_Model->getQuestionById($id);
    if($question != NULL) {
        $this->response(
            array(
                'status' => 1,
                'message' => 'Successfully displayed the question',
                'data' => $question
            ), REST_Controller::HTTP_OK
        );
    } else {
        $this->response(
            array(
                'status' => 0,
                'message' => 'There is no question with that id',
                'data' => $question
            ), REST_Controller::HTTP_NOT_FOUND
        );
    }
}
}

```

```

public function index_post() {
    $title = $this->security->xss_clean($this->post('title'));
    $description = $this->security->xss_clean($this->post('description'));
    $tags = $this->security->xss_clean($this->post('tags'));
    $user_id = $this->security->xss_clean($this->post('user_id'));

    $this->form_validation->set_rules('title', 'Title', 'required');
    $this->form_validation->set_rules('description', 'Description', 'required');
    $this->form_validation->set_rules('tags', 'Tags', 'required');
    $this->form_validation->set_rules('user_id', 'User ID', 'required');

    if($this->form_validation->run() == FALSE) {
        $this->response(
            array(
                "status" => 0,
                "message" => "Failed to validate form data"
            ),
            REST_Controller::HTTP_NOT_FOUND
        );
    } else {
        if(!empty($title) && !empty($description) && !empty($tags) && !empty($user_id)) {
            $question = array(
                'title' => $title,
                'description' => $description,
                'tags' => $tags,
                'user_id' => $user_id
            );
            if ($this->Questions_Model->createQuestion($question)) {
                $this->response(
                    array(
                        'status' => 1,
                        'message' => "Successfully added the question to the database"
                    ),
                    REST_Controller::HTTP_OK
                );
            } else {
                $this->response(
                    array(
                        'status' => 0,
                        'message' => "Failed to add the user to the database"
                    ),
                    REST_Controller::HTTP_INTERNAL_SERVER_ERROR
                );
            }
        } else {
            $this->response(

```

```

        array(
            "status" => 0,
            "message" => "All fields needs to be filled",
        ),
        REST_Controller::HTTP_NOT_FOUND
    );
}
}
}

public function answers_get() {
    $id = $this->input->get('id');
    $answers = $this->Questions_Model->listAnswersForAQuestion($id);
    if($answers > 0) {
        $this->response(
            array(
                'status' => 1,
                'message' => 'Successfully listed all answers',
                'data' => $answers
            ), REST_Controller::HTTP_OK
        );
    } else {
        $this->response(
            array(
                'status' => 0,
                'message' => 'There are no answers for this question',
                'data' => $answers
            ), REST_Controller::HTTP_NOT_FOUND
        );
    }
}

public function answers_post() {
    $comment = $this->security->xss_clean($this->post('comment'));
    $user_id = $this->security->xss_clean($this->post('user_id'));
    $question_id = $this->security->xss_clean($this->post('question_id'));

    $this->form_validation->set_rules('comment', 'Comment', 'required');
    $this->form_validation->set_rules('user_id', 'User ID', 'required');
    $this->form_validation->set_rules('question_id', 'Question ID', 'required');

    if($this->form_validation->run() == FALSE) {
        $this->response(
            array(
                "status" => 0,
                "message" => "Failed to validate form data"
            )
        );
    }
}
}

```

```

    ),
    REST_Controller::HTTP_NOT_FOUND
);
} else {
    if(!empty($comment) && !empty($user_id) && !empty($question_id)) {
        $answer = array(
            'comment' => $comment,
            'user_id' => $user_id,
            'question_id' => $question_id
        );
        if ($this->Questions_Model->createAnswer($answer)) {
            $this->response(
                array(
                    'status' => 1,
                    'message' => "Successfully added the answer to the database"
                ),
                REST_Controller::HTTP_OK
            );
        } else {
            $this->response(
                array(
                    'status' => 0,
                    'message' => "Failed to add the answer to the database"
                ),
                REST_Controller::HTTP_INTERNAL_SERVER_ERROR
            );
        }
    } else {
        $this->response(
            array(
                "status" => 0,
                "message" => "All fields needs to be filled",
            ),
            REST_Controller::HTTP_NOT_FOUND
        );
    }
}
}
}
}
}

```

api/Users.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

use Restserver\Libraries\REST_Controller;
require APPPATH . 'libraries/REST_Controller.php';

```

```
class Users extends REST_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->database();
        $this->load->model('Users_Model');
        $this->load->library('form_validation');
        $this->load->helper('security');
    }

    public function signup_post() {
        $first_name = $this->security->xss_clean($this->post('first_name'));
        $last_name = $this->security->xss_clean($this->post('last_name'));
        $username = $this->security->xss_clean($this->post('username'));
        $email = $this->security->xss_clean($this->post('email'));
        $password = $this->security->xss_clean($this->post('password'));

        $this->form_validation->set_rules('first_name', 'First Name', 'required');
        $this->form_validation->set_rules('last_name', 'Last Name', 'required');
        $this->form_validation->set_rules('username', 'Username', 'required|is_unique[user.username]');
        $this->form_validation->set_rules('email', 'Email', 'required|valid_email');
        $this->form_validation->set_rules('password', 'Password', 'required');

        // hashing the password
        $hashed_password = password_hash($password, PASSWORD_DEFAULT);

        // Checking for errors
        if ($this->form_validation->run() == FALSE) {
            $this->response(
                array(
                    "status" => 0,
                    "message" => "Failed to validate form data"
                ),
                REST_Controller::HTTP_NOT_FOUND
            );
        } else {
            if (!empty($first_name) && !empty($last_name) && !empty($username) && !empty($email) && !empty($password)) {
                // all values are available
                $user = array(
                    'first_name' => $first_name,
                    'last_name' => $last_name,
                    'username' => $username,
                    'email' => $email,
                    'password' => $hashed_password
                );
            }
        }
    }
}
```

```

);
if ($this->Users_Model->signupUser($user)) {
    $this->response(
        array(
            "status" => 1,
            "message" => "Successfully added the user to the database"
        ),
        REST_Controller::HTTP_OK
    );
} else {
    $this->response(
        array(
            "status" => 0,
            "message" => "Failed to add the user to the database"
        ),
        REST_Controller::HTTP_INTERNAL_SERVER_ERROR
    );
}
} else {
    // there are some empty fields
    $this->response(
        array(
            "status" => 0,
            "message" => "All fields needs to be filled",
        ),
        REST_Controller::HTTP_NOT_FOUND
    );
}
}
}

public function login_post() {
    $username = $this->security->xss_clean($this->post('username'));
    $password = $this->security->xss_clean($this->post('password'));

    $this->response($this->Users_Model->authenticate($username, $password));
}
}

```

Questions.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Questions extends CI_Controller {

    public function __construct() {

```

```

    parent::__construct();
    $this->load->database();
    $this->load->model('Questions_Model');
}

/**
 * Home page
 * All questions are listed in the home page
 * This is the default controller
 * https://w1761369.users.ecs.westminster.ac.uk/2019483/index.php
 * http://w1761369.users.ecs.westminster.ac.uk/2019483/index.php/questions/home
 * */
public function index() {
    $this->data['questions'] = $this->Questions_Model->listQuestions();
    $this->load->view('home-page', $this->data);
}

public function question() {
    $this->load->view('question-page');
}

public function addQuestion() {
    $this->load->view('add-question-page.php');
}

public function about() {
    $this->load->view('about-page');
}
}

```

Users.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Users extends CI_Controller {

    public function __construct() {
        parent::__construct();
    }

    /**
     * Login page
     * http://w1761369.users.ecs.westminster.ac.uk/2019483/index.php/user/login
     * */
    public function login() {
        $this->load->view('login-page');
    }
}

```

```

}

/**
 * Signup page
 * http://w1761369.users.ecs.westminster.ac.uk/2019483/index.php/user/signup
 */
public function signup() {
    $this->load->view('signup-page');
}
}

```

Answers.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Answers extends CI_Controller {

    public function __construct() {
        parent::__construct();
    }

    /**
     * Home page
     * All questions are listed in the home page
     * This is the default controller
     * https://w1761369.users.ecs.westminster.ac.uk/2019483/index.php
     * http://w1761369.users.ecs.westminster.ac.uk/2019483/index.php/questions/home
     */
    public function index() {
        $this->load->view('add-answer-page');
    }
}

```

Questions_Model.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Questions_Model extends CI_Model {

    // constructor
    public function __construct() {
        parent::__construct();
        $this->load->database();
    }
}

```



```

public function listQuestions() {
    $this->db->select('*');
    $this->db->from('question');
    $query = $this->db->get();
    return $query->result();

    $num_data_returned = $this->db->num_rows();

    if($num_data_returned < 1) {
        echo 'There is no data in the database';
        exit();
    }
}

public function getQuestionByID($id) {
    $this->db->where('id', $id);
    $query = $this->db->get('question');

    return $query->result();
}

public function createQuestion($data = array()) {
    return $this->db->insert("question", $data);
}

public function listAnswersForAQuestion($id) {
    $this->db->where('question_id', $id);
    $query = $this->db->get('answer');

    return $query->result();
}

public function createAnswer($data = array()) {
    return $this->db->insert("answer", $data);
}
}

```

Users_Model.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Users_Model extends CI_Model {
    function __construct()
    {
        parent::__construct();
        $this->load->database();
    }
}

```

```

}

public function signupUser($data = array()) {
    return $this->db->insert("user", $data);
}

public function authenticate($username, $password) {
    $this->db->where('username', $username);
    $query = $this->db->get('user');

    $result = $query->result();
    $num_rows = $query->num_rows();

    if ($num_rows > 0) {
        if (password_verify($password, $result[0]->password)) {
            return TRUE;
        } else {
            return FALSE;
        }
    } else {
        return FALSE;
    }
}
}

```

home-page.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests" />

    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/home-page-style.css">
    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/global-style.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Outlined" rel="stylesheet">

    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
    <script src="http://ajax.cdnjs.com/ajax/libs/underscore.js/1.3.3/underscore-min.js"></script>
    <script src="http://ajax.cdnjs.com/ajax/libs/backbone.js/0.9.2/backbone-min.js"></script>
</head>

<body>
    <!-- Navigation Bar -->

```



```

</div>

<script>
function viewAskQuestionPage() {
    window.location.assign("<?php echo base_url(); ?>index.php/Questions/addQuestion")
}
</script>
</body>

</html>

```

login-page.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/login-page-style.css">
    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/form-style.css" />
    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/global-style.css">
</head>

<body>
    <div class="container">
        <div class="info-section">
            
            <h1>Developer Support</h1>
        </div>
        <div class="vl"></div>
        <div class="form-setion">
            <div class="header">
                <h2>Login</h2>
            </div>
            <div class="input-section">
                <div class="form-group">
                    <label for="username">Username</label>
                    <input type="text" name="username" class="form-control">
                </div>
                <div class="form-group">
                    <label for="password">Password</label>
                    <input type="password" name="password" class="form-control">
                </div>
            </div>
            <div class="btn-panel-row">

```

```

        <button class="btn sign-up-btn">Sign Up</button>
        <div class="btn-panel-column">
            <div class="login-cancel-btns">
                <button class="btn cancel-btn">Cancel</button>
                <button class="btn login-btn">Login</button>
            </div>
            <a href="#">Forgot Password</a>
        </div>
    </div>
</div>
</div>
</body>

</html>

```

signup-page.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html>

    <head>
        <link rel='stylesheet' type='text/css' href="<?php echo base_url(); ?>public/css/global-style.css" />
        <link rel='stylesheet' type='text/css' href="<?php echo base_url(); ?>public/css/form-style.css" />
        <link rel='stylesheet' type='text/css' href="<?php echo base_url(); ?>public/css/signup-page-style.css" />
        <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Material+Symbols+Outlined:opsz,wght,FILL,GRAD@48,400,0,0" />
    </head>

    <body>
        <div class="container" id="signup-form">
            <div class="header">
                <button><img></button>
                <h2>Sign Up</h2>
            </div>
            <div class="form-section">
                <div class="left">
                    <div class="form-group">
                        <label for="firstName">First Name</label>
                        <input id="firstName" type="text" name="firstName" class="form-control">
                    </div>
                    <div class="form-group">
                        <label for="username">Username</label>
                        <input id="username" type="text" name="username" class="form-control">

```

```

    </div>
    <div class="form-group">
        <label for="password">Password</label>
        <input id="password" type="password" name="password" class="form-control">
    </div>
</div>
<div class="right">
    <div class="form-group">
        <label for="lastName">Last Name</label>
        <input id="lastName" type="text" name="lastName" class="form-control">
    </div>
    <div class="form-group">
        <label for="email">Email</label>
        <input id="email" type="text" name="email" class="form-control">
    </div>
    <div class="form-group">
        <label for="confPassword">Confirm Password</label>
        <input type="password" name="confPasswrod" class="form-control">
    </div>
</div>
</div>
<div class="btn-section">
    <button id="signup-btn" class="btn sign-up-btn">Sign Up</button>
</div>
</div>

<script>
var User = Backbone.Model.extend({
    uri: "<php echo base_url(); ?>api/users/signup",
    defaults: {
        first_name: null,
        last_name: null,
        username: null,
        email: null,
        password: null
    },
    isValid: function() {
        if (this.get('first_name') == "") {
            alert('First Name field is empty.')
            return FALSE;
        } else if (this.get('last_name') == "") {
            alert('Last Name field is empty.')
            return FALSE;
        } else if (this.get('username') == "") {
            alert('Username field is empty.')
            return FALSE;
        }
    }
});

```

```

    } else if (this.get('email') == "") {
        alert('Email field is empty.')
        return FALSE;
    } else if (this.get('password') == "") {
        alert('Password field is empty.')
        return FALSE;
    } else {
        return TRUE;
    }
},
});

var Signup = Backbone.View.extend({
    model: User,
    events: {
        'click #signup-btn': 'signup'
    },
    signup: function(e) {
        e.preventDefault();

        var user = new User({
            first_name: $('#firstName').val(),
            last_name: $('#lastName').val(),
            username: $('#username').val(),
            email: $('#email').val(),
            password: $('#password').val()
        });

        if (user.isValid()) {
            user.save({}, {
                async: FALSE,
                contentType: 'application/json',
                dataType: 'text',
                success: function(model, response) {
                    alert('Succesfully created the User');
                    window.location.href =
                        "<?php echo base_url(); ?>index.php/users/signup";
                }
            });
        }
    },
});

var signup = new Signup({
    el: $('#signup-form')
});

```

```
</script>

</body>

</html>
```

add-question-page.php

```
<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>

<!DOCTYPE html>
<html lang="en">

    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests" />

        <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/add-question-page-
style.css">
        <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/global-style.css">
        <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Outlined" rel="stylesheet">

        <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
        <script src="http://ajax.cdnjs.com/ajax/libs/underscore.js/1.3.3/underscore-min.js"></script>
        <script src="http://ajax.cdnjs.com/ajax/libs/backbone.js/0.9.2/backbone-min.js"></script>
    </head>

    <body>
        <div class="main-container">
            <div class="header">
                <h1>Add Question</h1>
            </div>
            <div class="text-section">
                <label for="question"></label>
                <textarea class="question-title" placeholder="Type you question title here"></textarea>
                <textarea class="question-desc" placeholder="Type you question description here"></textarea>
            </div>
            <div class="tag-section">
                <h2>Add Tags</h2>
                <div class="input-section">
                    <input type="text" name="tag" placeholder="#CodeIgniter3" />
                    <input type="text" name="tag" placeholder="#CodeIgniter3" />
                    <input type="text" name="tag" placeholder="#CodeIgniter3" />
                    <input type="text" name="tag" placeholder="#CodeIgniter3" />
                    <input type="text" name="tag" placeholder="#CodeIgniter3" />
                </div>
            </div>
        </div>
    </body>
</html>
```



```

        </div>
        <button class="add-question-btn">Add Question</button>
    </div>
</body>

</html>

```

add-answer-page.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>

<!DOCTYPE html>
<html lang="en">

    <head>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests" />

        <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/add-answer-page-
style.css">
        <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/global-style.css">
        <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Outlined" rel="stylesheet">

        <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
        <script src="http://ajax.cdnjs.com/ajax/libs/underscore.js/1.3.3/underscore-min.js"></script>
        <script src="http://ajax.cdnjs.com/ajax/libs/backbone.js/0.9.2/backbone-min.js"></script>
    </head>

    <body>
        <div class="main-container">
            <div class="header">
                <h1>Add Answer</h1>
            </div>
            <div class="text-section">
                <label for="answer"></label>
                <textarea class="answer" placeholder="Type you answer here"></textarea>
            </div>
            <button class="add-answer-btn">Add Answer</button>
        </div>
    </body>

</html>

```

nav-bar-widiget.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>

```

```

<!DOCTYPE html>
<html lang="en">

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

    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/navbar-style.css">
    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/global-style.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Outlined" rel="stylesheet">
</head>

<body>
    <div class="navbar">
        <div class="leader">
            
            <h2>Developer Support</h2>
            <button><img></button>
        </div>
        <div class="navbar-items">
            <div class="nav-bar-item-card">
                <p><a href="<?php echo base_url(); ?>index.php/questions/index">Home</a></p>
            </div>
            <div class="nav-bar-item-card">
                <p><a href="<?php echo base_url(); ?>index.php/questions/about">About</a></p>
            </div>
        </div>
    </div>
</body>

</html>

```

question-page.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests" />

    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/question-page-style.css">
    <link rel="stylesheet" type="text/css" href="<?php echo base_url(); ?>public/css/global-style.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons+Outlined" rel="stylesheet">

```

```

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
<script src="http://ajax.cdnjs.com/ajax/libs/underscore.js/1.3.3/underscore-min.js"></script>
<script src="http://ajax.cdnjs.com/ajax/libs/backbone.js/0.9.2/backbone-min.js"></script>
</head>

<body>
  <!-- Navigation Bar -->
  <?php include 'application/views/nav-bar-widget.php';?>

  <div class="body">
    <div class="header">
      <pre class="date"><strong>Date:  </strong>15/01/2023</pre>
      <div class="question-area">
        <h3>Why does kotlin/native hello world not output anything inside IntelliJ IDEA?</h3>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mauris magna, elementum vel
          aliquam eu, pretium gravida est. Nam eu ante at massa luctus eleifend. Sed fringilla justo nec
          diam consectetur, et suscipit magna sagittis. Vestibulum vulputate, metus sit amet mollis
          consectetur, ex nisi efficitur lacus, a faucibus purus metus vitae orci. Sed sit amet rhoncus
          ante, at consequat orci. Vivamus pellentesque vel enim vitae ullamcorper. Mauris consectetur
          lacinia velit non dictum. Suspendisse nec suscipit ipsum. Vestibulum rhoncus mi scelerisque eros
          posuere, vel efficitur neque blandit. Suspendisse vestibulum purus vitae dui tempor accumsan.
          Aliquam id est justo. Nullam rhoncus a erat a molestie. Suspendisse eleifend diam sed tincidunt
          egestas. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis
          egestas. Aliquam erat volutpat.</p>
        </div>
        <div class="btn-tag-panel">
          <button class="answer-btn">Answer</button>
          <div class="like-btn"><button><img></button>
            <p>30</p>
          </div>
          <div class="dislike-btn"><button><img></button>
            <p>15</p>
          </div>
        </div>
      </div>

      <div class="main-container">
        <?php for ($x = 0; $x <= 10; $x++) { ?>
          <div class="answer-card">
            <div class="rating-section">
              <div class="like-btn">
                <button><img></button>
                <p>30</p>
              </div>
              <div class="dislike-btn">

```

```

        <button><img></button>
        <p>10</p>
    </div>
</div>
<div class="vl"></div>
<div class="main-section">
    <p>Why does kotlin/native hello world not output anything inside IntelliJ
        IDEA?</p>
</div>
<div class="right-section">
    <div class="date">
        <p>14/01/2023</p>
    </div>
</div>
</div>
<?php } ?>
</div>
</div>

<!-- <script>
$(document).ready(function() {
    $('#answer-section').click(function(event) {
        event.preventDefault();
        console.log(this.className);
        if (this.className == 'answer-card') {
            $('.answer-card').addClass('answer-card-extend');
            $('#p-extend').addClass('p-extend');
        } else {
            $('.answer-card').removeClass('answer-card-extend');
            $('#p-extend').removeClass('p-extend');
        }
    })
})
</script> -->
</body>

</html>

```

w1761369_0.sql

```

-- --
-- -- Database: `w1761369_0`
-- --

CREATE TABLE `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,

```

```

`first_name` varchar(255) NOT NULL,
`last_name` varchar(255) NOT NULL,
`username` varchar(255) NOT NULL,
`email` varchar(255) NOT NULL,
`password` varchar(255) NOT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `user` (`id`, `first_name`, `last_name`, `username`, `email`, `password`, `created_at`) VALUES
(1, 'Shawn', 'Marley', 'sm1995', 'sh@email.com', 'sh123@123', '2023-01-04 10:58:28'),
(2, 'Gomey', 'Khan', 'gk1999', 'gk@email.com', 'gk123@123', '2023-01-04 10:58:28'),
(3, 'Thaveesha', 'Dheeraseskera', 'thaveesha', 'thaveesha.00@email.com',
'$2y$10$cMN0Fz1r5UMGSHT..uAYgu.KCSJ/0uAFzulFZr.lxRjk8qD4BafPK', '2023-01-15 01:05:39'),
(4, 'Shanuka', 'Peiris', 'shanuka', 'shanuka.00@email.com',
'$2y$10$GVs04vHiRpl3P.YanpDyfumg2hxygd4EjipyATGgBkF/QZcqftbhW', '2023-01-15 01:07:07');

CREATE TABLE `question` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`title` mediumtext NOT NULL,
`description` longtext NOT NULL,
`tags` text,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`user_id` int(11) NOT NULL,
PRIMARY KEY(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `question` (`id`, `title`, `description`, `tags`, `created_at`, `user_id`) VALUES
(1, 'What is \'foo\' is programming?', 'What is \'foo\' is programming?\nCan someone explain this...', NULL,
'2023-01-04 11:01:57', 1),
(2, 'How do I fetch data from a database by ID in CodeIgniter?', 'How do I fetch data from a database by ID in
CodeIgniter?\nCan someone explain this...', 'CodeIgniter,Database', '2023-01-15 04:15:32', 2);

CREATE TABLE `answer` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`comment` longtext DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`user_id` int(11) NOT NULL,
`question_id` int(11) NOT NULL,
PRIMARY KEY(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

INSERT INTO `answer` (`id`, `comment`, `created_at`, `user_id`, `question_id`) VALUES
(1, 'It is a placeholder for a function/method\r\nexample: \r\n class foo {\r\n // code\r\n }\r\n', '2023-01-04
12:21:10', 1, 1);

```

```

CREATE TABLE `question_rating` (
  `id` int(11) NOT NULL,
  `rating` tinyint(1) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `user_id` int(11) NOT NULL,
  `question_id` int(11) NOT NULL,
  PRIMARY KEY(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE `answer_rating` (
  `id` int(11) NOT NULL,
  `rating` tinyint(1) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `user_id` int(11) NOT NULL,
  `answer_id` int(11) NOT NULL,
  PRIMARY KEY(`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

ALTER TABLE `question`
  ADD CONSTRAINT `fk_user_question` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`);

ALTER TABLE `answer`
  ADD CONSTRAINT `fk_question_answer` FOREIGN KEY (`question_id`) REFERENCES `question` (`id`),
  ADD CONSTRAINT `fk_user_answer` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`);

ALTER TABLE `question_rating`
  ADD CONSTRAINT `fk_question_rating` FOREIGN KEY (`question_id`) REFERENCES `question` (`id`),
  ADD CONSTRAINT `fk_user_question_rating` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`);

ALTER TABLE `answer_rating`
  ADD CONSTRAINT `fk_answer_rating` FOREIGN KEY (`answer_id`) REFERENCES `answer` (`id`),
  ADD CONSTRAINT `fk_user_answer_rating` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`);

```

2.

- Added a new table to add ratings for a question named 'question_rating' which has a primary key: 'id' and two foreign keys: 'user_id', 'question_id'.
- Added a new table named 'answer_rating' to add ratings for an answer which has a primary key: 'id' and two foreign keys: 'user_id', 'answer_id'.
- Created 'api/Users.php', 'api/Users.php to manage REST API endpoints and there are three controllers named 'Questions', 'Answers', 'Users' to load views in 'application/controllers' folder.
- Added a method in 'api/Users.php' controller to check whether an username is already in the database.

- Added methods to 'Questions_Model' model and 'Questions' controller to accommodate following functionalities,
 - Get all questions in the database.
 - Create a new question
 - Get a question using question id
 - Get all answers for a question
 - Create an answer for a question
 - Add rating to a question
 - Add rating to an answer

In my opinion, the changes made to the UI, database, controllers, models, etc., were useful in the development phase. Adding a new table for question ratings and answer ratings allows for more accurate tracking of user engagement and feedback. The creation of REST API endpoints and controllers for managing them, enables easy access and management of the data. Having a method to check for unique usernames in the database is an important feature for user registration.

The methods added to the 'Questions_Model' model and 'Questions' controller provide functionality for getting all questions and answers in the database, creating new questions and answers, adding ratings to questions and answers, and getting a question using question id. These functionalities are important for the smooth functioning of a question-answer platform.

If I were to start again, I would have paid more attention to the naming conventions of the tables, controllers and models. I would also have made sure that the controllers and models are in a more modular structure which would have made it easier to scale in the future.

3. Q1:

The first reason the author provides for not using web frameworks is that they add unnecessary complexity to the codebase. The author argues that frameworks often come with a lot of features that may not be needed for a specific project, but still need to be understood and navigated by developers. This can make it harder for new developers to understand and work on the code and can also lead to increased maintenance costs.

The second reason the author provides is that frameworks can limit flexibility. The author argues that by using a framework, developers are forced to conform to its conventions and patterns, which can make it difficult to customize the application to specific needs. Additionally, frameworks can make it harder to switch to a different technology stack in the future.

The third reason the author provides is that frameworks can lead to a decrease in performance. The author argues that the overhead of the framework can slow down the application, and that the abstraction provided by the framework can make it harder to optimize specific parts of the application for performance.

The author supports these reasons by citing examples from his personal experience of working with various web frameworks and encountering these issues. He also mentions that frameworks can have their use cases, but it's important to consider if the particular use case is worth the trade-offs of using a framework. The author suggests that it may be better to use a minimalistic, un-opinionated library or build the application from scratch if the project is small and simple, in order to avoid the complexity, inflexibility and performance issues that frameworks can bring.

3. Q2:

The most important reason the author gives against using web frameworks is that they add unnecessary complexity to the codebase. The author argues that frameworks often come with a lot of features that may not be needed for a specific project, but still need to be understood and navigated by developers. This can make it harder for new developers to understand and work on the code and can also lead to increased maintenance costs.

I agree with the author's argument that frameworks can add unnecessary complexity to the codebase. The use of a framework can often lead to a larger codebase that is more difficult to navigate and understand, which can make it harder for new developers to get up to speed and can also lead to increased maintenance costs. Additionally, the use of a framework can often lead to a lot of boilerplate code that may not be needed for the specific project, which can also add unnecessary complexity.

However, I also believe that frameworks can have their use cases and can be beneficial in certain situations. For example, frameworks can provide a lot of useful functionality out of the box and can make it easier to develop an application quickly. Additionally, frameworks can provide a consistent structure and a set of conventions which can make it easier for developers to work together and can also make it easier to maintain the codebase.

In conclusion, I agree with the author that frameworks can add unnecessary complexity to the codebase, but I also believe that frameworks can have their use cases, depending on the specific project and the needs of the development team. It's important for developers to weigh the pros and cons of using a framework and to decide if the added complexity is worth it for the specific use case.