**Project**: Implementation of a digital library search engine with a chatbot for electronic theses and dissertations.

**Student name**: Kumushini Thennakoon

**Student UIN**: 01251889

**All sections are mandatory. Please do not change the format of this template.**

1. **Overview**

The web site is a digital library search engine enabled with a chatbot option containing electronic thesis and dissertations. I’m using PHP for developing the website interface and its functions, MySQL for the database operations, and running on Apache server. I have not used any frameworks to build the website.

1. **Milestone Accomplishments**

**List ALL specifications of this milestone and specify whether certain specifications are fulfilled or not.**

Table 1: Status of milestone specifications.

|  |  |  |
| --- | --- | --- |
| **Fulfilled** | **Feature#** | **Specification** |
| Yes | 1 | Users should be able to register new accounts using email addresses. |
| Yes | 2 | Users are identified by email address. |
| Yes | 3 | Password must be encrypted before storing in the database. |
| Yes | 4 | Users cannot register duplicate accounts using the same email address. |
| Yes | 5 | An admin user should be created from the backend. |
| Yes | 6 | Users cannot log in to the system until their requests are approved by the admin. |
| Yes | 7 | An admin user has a different view from a regular user: it can approve users (for now). |
| Yes | 8 | Users should be able to log into your website using the accounts they registered. |
| Yes | 9 | Users should be able to reset their passwords if they forget it. |
| Yes | 10 | Users should be able to change their passwords after they login. |
| Yes | 11 | A 2-factor-authentication should be used when a user attempt to login. This can be done by email, phone text, or a DUO push. You can just implement one of them. |
| Yes | 12 | The website should have a homepage for each user, where they can view their profiles, change passwords, and update information. Email cannot not be changed. |
| Yes | 13 | The website should provide a search box at the landing page (without logging in). The searching function may not be working at this stage, and there should be a search button next to the search box. |

1. **Architecture**

**Describe the overall architecture of this project, including but not limited to the programming language, framework used, the major components and how they are connected. Draw a diagram similar to the one below.**

**A diagram of a scientific experiment

Description automatically generated with medium confidence**

In-progress

1. **Database Design**

Describe your database design including a table showing fields, data types, key types, and a few examples of rows.

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Key | Example |
| email | VARCHAR(100) | Primary | joshallen@gmail.com |
| first\_name | VARCHAR(50) |  | John |
| last\_name | VARCHAR(50) |  | Allen |
| password | VARCHAR(255) |  | \*\*\*\*\*\* |
| user\_approved\_date | DATETIME |  | 2023.09.28 |
| user\_signed\_up\_date | DATETIME |  | 2023.09.27 |
| user\_role | ENUM (‘user’,’admin’) |  | User |
| user\_status | ENUM (‘approved’,’signed\_up’,’manually\_added’) |  | approved |
| tmp\_pwd | ENUM (‘yes’,’no’) |  | no |
| g2fa\_key | VARCHAR(300) |  |  |

1. **Implementation**

Describe how **each** specification is implemented. **Specify and highlight the program file in which each specification is implemented**. An example is shown below.

*Search box and search button (in progress):*

The search box at the landing page of my website allows users to search materials on the website. A user can type the words in the search box and click search button.

*Navigation bar (in progress):*

Navigation bar is included in all the views of the website. The “Archiveia” item and “Home” item all ways direct users to the index page (index.php). Code : src 🡪 navbar.php

*Sign up button:*

The sign up button is in the index page of the website (index.php). When a user clicks the sign up button it directs to a form asking user to enter first name, last name, email(username), password, and confirm password. Then user can submit the form. After submission the data will be stored in the database according to the insert query in the signup.php file and directs to the login page. The password will be encrypted with password\_hash() function before storing in the database. Code : src 🡪 signup.php

*Reset button in sign up form:*

This button clears the user input fields.

*Login button:*

Login button is in the index page (index.php) and a user who is already signed up and approved by admin can log into the website using login button. When the login button is clicked it directs the user to the login form which ask user to enter username(email) and password. There are two kinds of users as admin approved(approved) users and admin not yet approved (signed up) users. If a user fill the login form and click submit button, it checks user\_status, if the user-status is signed\_up, an error message is popped as “Please wait until admin approval”. If user\_status is approved, login.php file validate the user input and directs to two factor verification process and take user to the user account. Code: src 🡪 login.php

*Two factor verification:*

When a user (user\_status = approved) tried to login with login form, after form submission a verification code is generated in google2fa\_test.php and send as an email to the user email using send\_mail\_2fa.php script. User is directed to a new form requesting the verification code received. User enters the verification code and submit, after the user\_role (admin, user) is checked it directs user to the relevant account. Code: src 🡪 google2fa\_test.php, send\_mail\_2fa.php

*Forgot password link:*

Forgot password link is included in the login form. If a user cannot remember password, he/she can click forgot password link to reset the password. This link direct user to a new form named Reset password link and ask user to enter the email address (username). When the user enters the email address and click submit button, a six-digit temporary password is generated in reset\_password\_link.php and sent to the entered email address with send\_mail\_reset\_password.php and take user to a new change password form in change\_pwd\_with\_reset\_pwd.php. The form takes the temporary password received via email as user input, new password, and confirm new password. The user entered temporary password is verified with already stored value in the data base in temp\_pwd column. I there are no errors user is directed to the login page. Code: src 🡪 reset\_password\_link.php, send\_mail\_reset\_password.php, change\_pwd\_with\_reset\_pwd.php.

*Admin view:*

An admin can log into the web site using the email address and password. Admin also undergo the two-step verification when logging in. Admin (admin.php) has a different view than a regular user. He can view profile information and all the users signed into the website. He also can approve users to “approved” status with a button click. Change password, update information, and sign out functions are available for the admin.

*Approve user:*

After loggin admin can approve users using approve button. When admin click the approve button, approve\_user.php file start to execute and the update query inside the php file updates the database and change the user\_status to “approved”. Code: src 🡪 approve\_user.php

*User View:*

An approved user has an account and it contain profile information, change password, update information, and sign out options.

*Change password:*

A regular user or an admin can change password using change password button. This button is in both accounts. When the button is clicked user will be directed to the change password form. Change password form contains three user inputs as current password, new password, and confirm new password. After getting user input change\_password.php file verifies the current password and update new password (encrypted) inside the database with an update query. Code: src 🡪 change\_password.php

*Cancel link in change password form:*

This link will direct user to the login page without changing password.

*Update information:*

A regular user or an admin can update information with update information button. This function allows users to only update the first name and last name. With the button click user will be directed to fill the update information form. Inside the form first name and last name is taken as user inputs and runs an update query inside the update\_info.php file. Code: src 🡪 update\_info.php

*Cancel link in Update information form:*

This link will take user to the user account without updating information.

*Sign out:*

A regular user or an admin can sign out from the account by clicking “Sign out of your account” button. After clicking the button logout.php file will execute and destroys the session. User will be directed to the login page. Code: src 🡪 logout.php