**Project**: Digital library search engine with a Chatbot

**Student name**: Maaz Hasnain Khan

**Student UIN**: 01244465

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

1. **Overview**

The website we are working on is a Digital Library search engine with a Chatbot. We are using Create React App (CRA) Framework and Node.js API. For this milestone we are designing a login & registration page with the features mentioned below:

1. **Milestone Accomplishments**

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

Table : Status of milestone specifications.

|  |  |  |
| --- | --- | --- |
| **Fulfilled** | **Feature#** | **Specification** |
| Yes | 1 | 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. |
| 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. |
| 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**

The architecture follows a standard web application model with a client-server interaction pattern, where Create React App (CRA) serves as the client (frontend) and Node.js as the server (backend). The database stores and retrieves user data, and various pages and components provide different functionalities as described in your project.

A diagram of a computer program

Description automatically generated

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 |
| user\_id | INT | Primary autoincremented | 1 |
| name | TEXT |  | Johnsmith |
| email | TEXT |  | ODU |
| password | TEXT |  | 1990 |
| guid | VARCHAR(100) |  | 04811866-1952-4f01 |
| is\_admin | INT |  | 0 |
| active\_user | INT |  | 0 |
| otp | VARCHAR(4) |  | 4567 |

1. **Implementation**

*Search box and search button:*

The search box and search button is at the landing page of my website. The search bar and button are not working at the moment. The code for this page is insearch-bar.tsx

*Login:*

The login page can be accessed through the header. It contains text box for inputting the email address and password, which if found correct, will direct the user to the OTP page. There is also a button on this page which is required to be pushed once the user inputs his or her email and password to be redirected. The code for this page can be found in login.tsx

*Signup:*

The sign-up page can be accessed through the header. It contains text box where user will enter required information for registration of an account. The password will be stored in database in encrypted form. Also, the user won’t be able to register duplicate account using the same email address. There is also a button at the end of the form which has to be pushed to complete the registration. The code for this page can be found in sign-up.tsx

*Forget Password:*

The forget password page can be accessed through the login page. It contains a text box where user will enter his or her email address which will send a URL for resetting the password at that email address. There is also a button which will complete this request. The code for this page can be found in forgot-password.tsx

*Reset Password:*

The reset password page can be accessed once the user has opened the link found in his email after registering for forget password application. It contains text boxes where user will enter the new password. This password will be replaced by the old password in the database. The code can be found in reset-password.tsx

*OTP:*

The OTP page is accessed when the user successfully logs into his or her account. The user will receive an email with the OTP which he has to enter in the OTP page. If the authentication is successful, the user will be directed to the profile. The code for this page can be found in otp.tsx

*Admin Page:*

When the admin logs in, instead of the profile page he will land at a page where he can see a table of registered users. He will have an option of activating and deactivating the user’s account. Without the admin’s approval the users will not be able to log in to their account. The code for this page can be found in user-admin.tsx

*Profile:*

When a user logs in successfully, he will be directed to the profile page. In the profile page his email address and name will be displayed along with a text box of password. He can change his password and name but not his email address. The code for this page can be found in profile.tsx