**Design Document**

**Table of Contents**

1. Introduction
2. User Interface
3. Data Flow diagram
4. Login Page
5. Registration Page
6. Home Page
7. PostGreSql
8. **Introduction**

**1.1. Description of the project**

This is simple login page created by using frontend technology. Where user can enter there credential and click on log in button.

**1.2. Required Software**

a) Express

b) VS Code

c) Postgre SQL

d) node Js

e) Knex

f) nodemon

**1.3. Document Scope**

This document will cover the technical design details related to creating UI, frontend, and integration with the website.

**1.4 Required Technology:-**

a) HTML-5

b) Bootstrap-5

c) JavaScript

d) CSS

1. **User Interface**

**2.1 HTML-5:-**

Hypertext markup language or HTML is the standard markup language for documents designed to be displayed in a web browser.

**2.2 CSS :-**

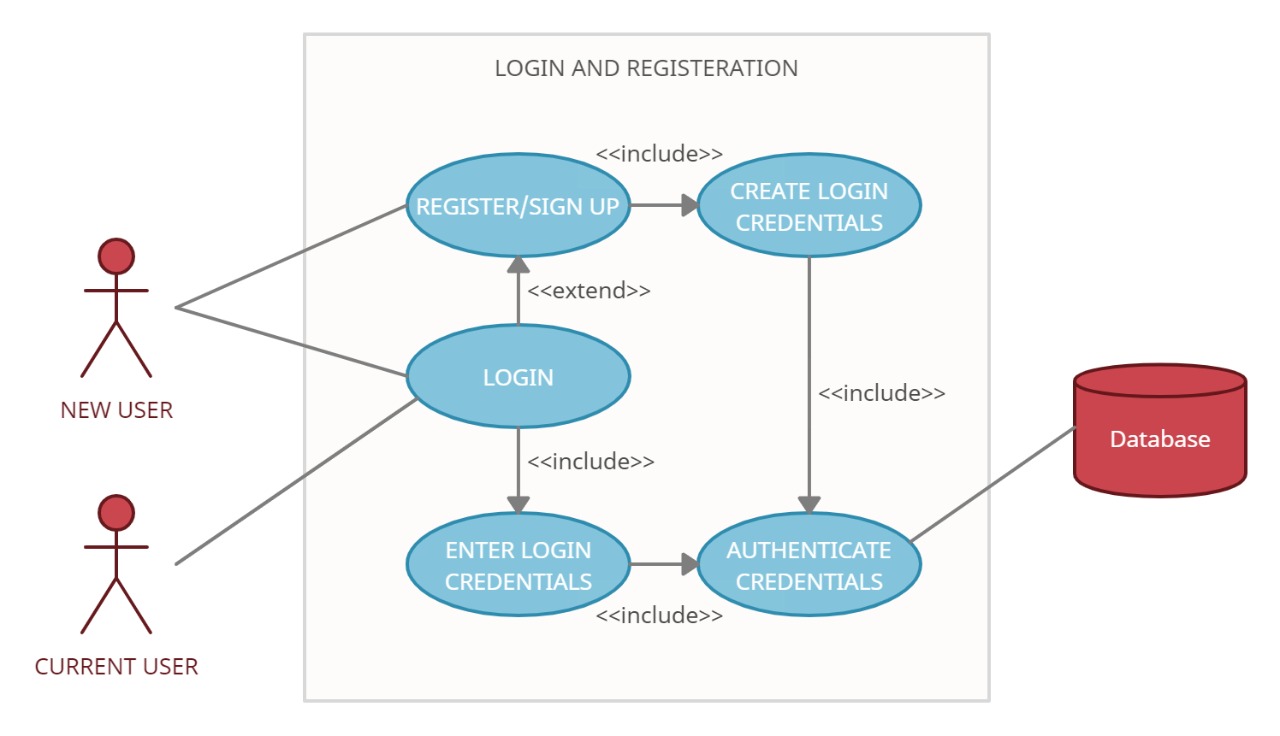
CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

**2.3 Bootstrap-5:-**

Bootstrap 5 is the newest version of Bootstrap, which is the most popular HTML, CSS, and JavaScript framework for creating responsive, mobile-first websites.

**3.Data Flow Diagram**

**Login and Registration:-**



|  |  |
| --- | --- |
| Brief Description | Login and Registration |
| Basic Flow | This use case describes how a user logs in or register in to the system   1. The user has to register himself into the system. 2. After the successful registration, student types his/her name and password on the login form. 3. The system validates the student’s password and logs him/her into the system. 4. The User can see the entire application/ Home Page. 5. When user press Logout Button the page will redirect to Login page.  * Password * Email Id |
| Alternate Flow | 1. If the system can find the textbox field empty or invalid, then it displays the error message on the screen. |
| Validation | 1. The Name shouldn’t be empty. 2. The Email should be letters with @ and . it should not be empty. 3. The Password should not be empty. And must contain atleast 6 letters or numbers. |
| Pre-Conditions | User should have network access and Browser with latest updates. |
| Post-Conditions | Home Page will be shown. |

**Wireframe for Login:**

**Log In Page**

|  |  |
| --- | --- |
| **Username** | textbox |
| **Password** | textbox |

****

**Registration Page**

|  |  |
| --- | --- |
| **Name** | textbox |
| **Email** | textbox |
| **Password** | textbox |

Submit

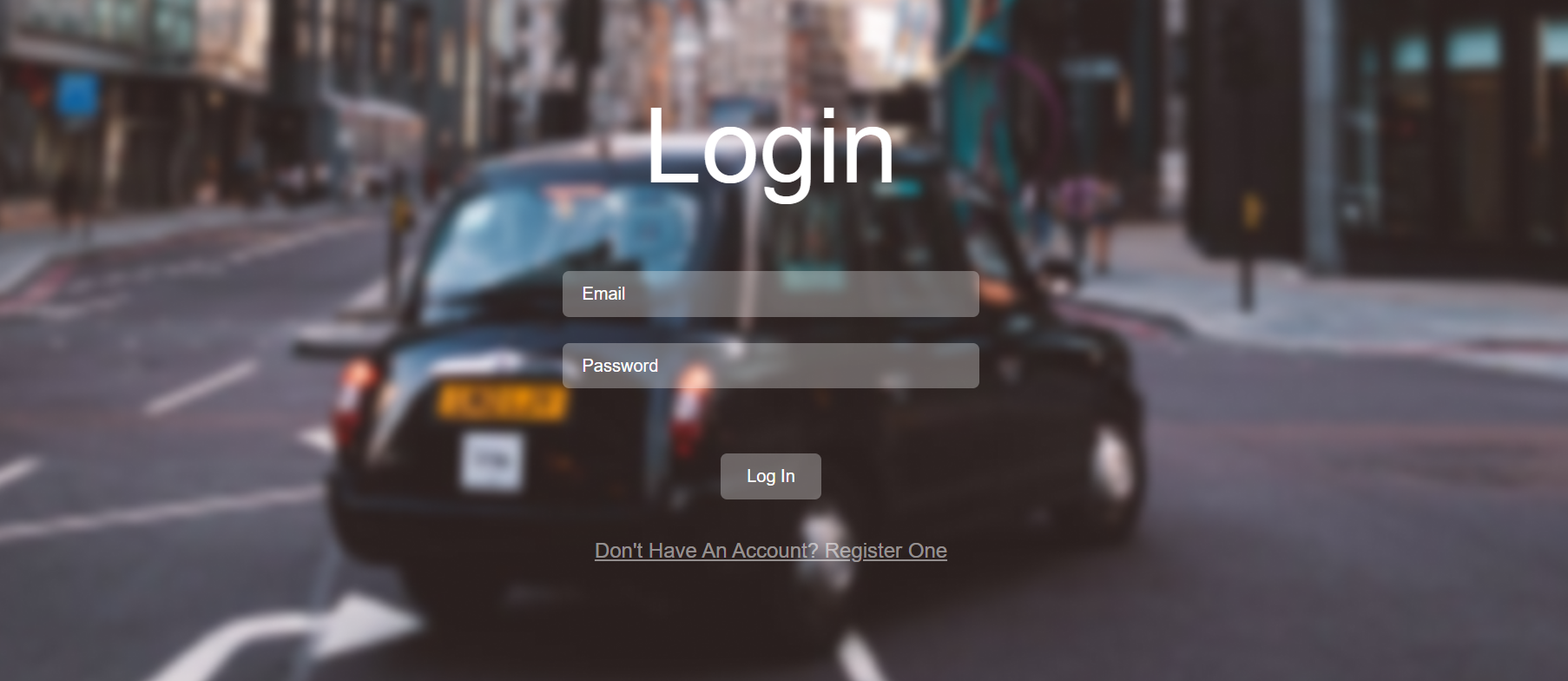
1. **Login Page**

This is the sign in page of our application here we have used two textboxes to get user information for authentication.

• In center of page user can see one textbox to fill there username/Email.

• In below of email/username user can see one more textbox to make their password.

• In bottom of the textbox’s user can see one log in button When user click on log in button it will take user in to the application’s home page.

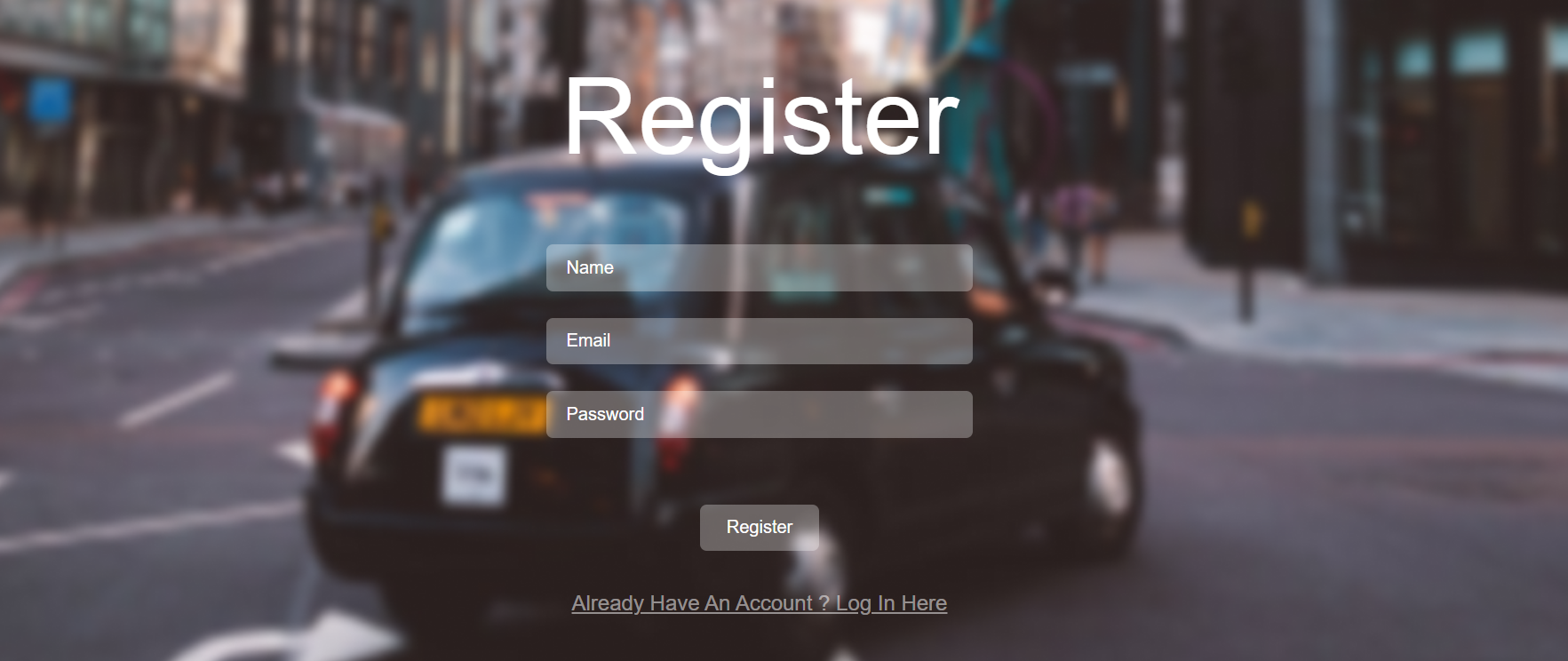


* If user don’t have an account in the application then user can click on the link given below for registration or creating the account.

1. **Register Page**

In this page user can register or create their account in the application.

* In first textbox user need to enter their name.
* In second textbox user need to give their email address.
* In third textbox user should give their password.
* And then user can click on Register button.

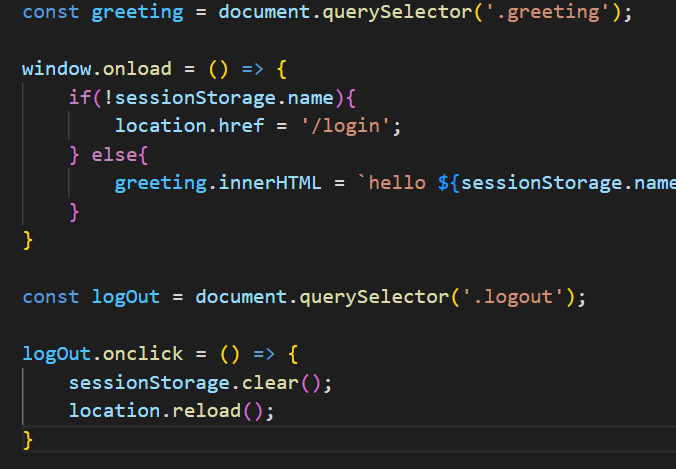
****

* After clicking on register button user can see the home page of website.

1. **Home Page**

In home page user can see their name that is fetched by database with greeting.



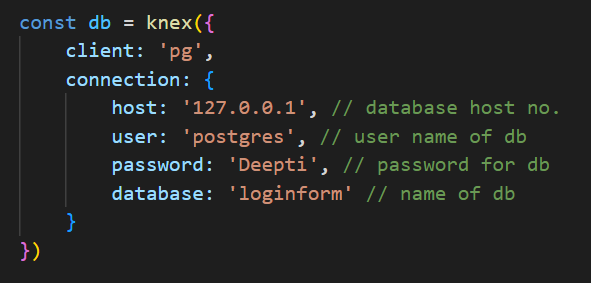


1. **Postgre SQL**

For storing user information I have used Postgre sql as a database.

To connect my frontend application with backend I have used knex.

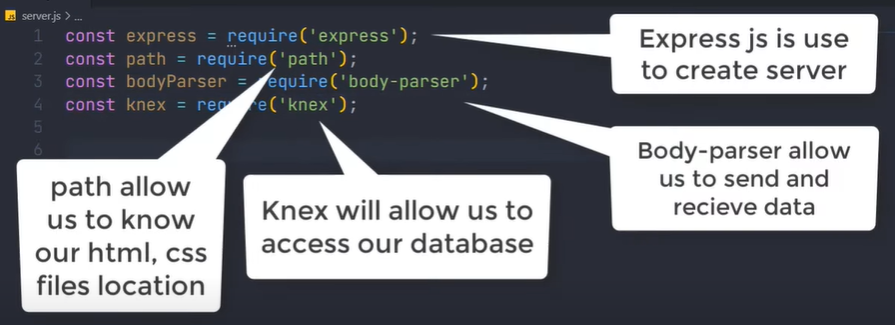
This is the code snippet:-



**Knex :-**

Knexis a SQL query builder, mainly used for Node. js applications with built in model schema creation, table migrations, connection pooling and seeding.

By using these credential we can connect our application with database.

****