Mess Bill Calculation System

Student Mess Bill Management System

# Project Contents

## Overview

The Student Mess Bill Management System is a web-based application aimed at automating and simplifying the process of mess billing, student attendance tracking, and expense calculation within hostels or educational institutions. It allows both students and administrators to interact with the system securely and effectively.

## Objective

To provide an efficient, transparent, and digital solution for managing student mess expenses and records.

## Users

* Admin: Manages student details, expenses, and attendance.
* Student: Views personal expenses and attendance.

## Functionalities

* Student Registration and Login
* Admin Login
* Monthly Expense Entry
* Attendance Tracking
* Expense Calculation
* Student Expense View
* Database Integration
* Secure Authentication
* User Interface for Admin and Students

# Project Codes

## 1. HTML Frontend Files

### index.html

* Landing page
* Links to Admin and Student login pages
* Clean layout with bootstrap for responsiveness

### studentlogin.html

* Login form for students
* Fields: Student ID, Phone Number
* Form validation with JavaScript

### studentregistration.html

* Registration form for new students
* Fields: ID, Name, Branch, Phone Number
* POST method used to submit data to backend

### studentexpenses.html

* Dashboard for students to view expenses
* Displays personal mess bills and attendance
* Conditional rendering for months and records

## 2. Backend (Flask)

* Python Flask app with route management
* @app.route decorators handle different functionalities
* Uses Flask sessions for user authentication
* Communicates with MySQL database using pymysql
* Data submitted via HTML forms is stored and retrieved dynamically

## 3. SQL File: student\_expense\_db1.sql

* Creates tables:
* student
* admin
* attendance
* expense
* student\_expense
* Populates data for initial use
* Enforces foreign key relationships

# Key Technologies

|  |  |
| --- | --- |
| **Layer** | **Technologies Used** |
| Frontend | HTML5, CSS3, JavaScript, Bootstrap |
| Backend | Python (Flask Framework) |
| Database | MySQL |
| Tools | Google Fonts, VS Code, Postman |
| Hosting | Localhost (XAMPP/WAMP) or Cloud Platforms |

# Description

## Admin Features

* Login: Secure authentication for backend access.
* Student Management: Register, update, or delete student details.
* Expense Entry: Record monthly mess expense.
* Attendance Entry: Input monthly attendance of students.
* Expense Calculation: Automatically divide total expense among students based on attendance.
* Dashboard: Overview of current statistics and control panel for managing features.

## Student Features

* Login: Secure access using ID and phone number.
* Expense History: View detailed mess bill per month.
* Attendance View: See attendance that affects billing.
* Profile: Personal details shown in dashboard

## Database Tables

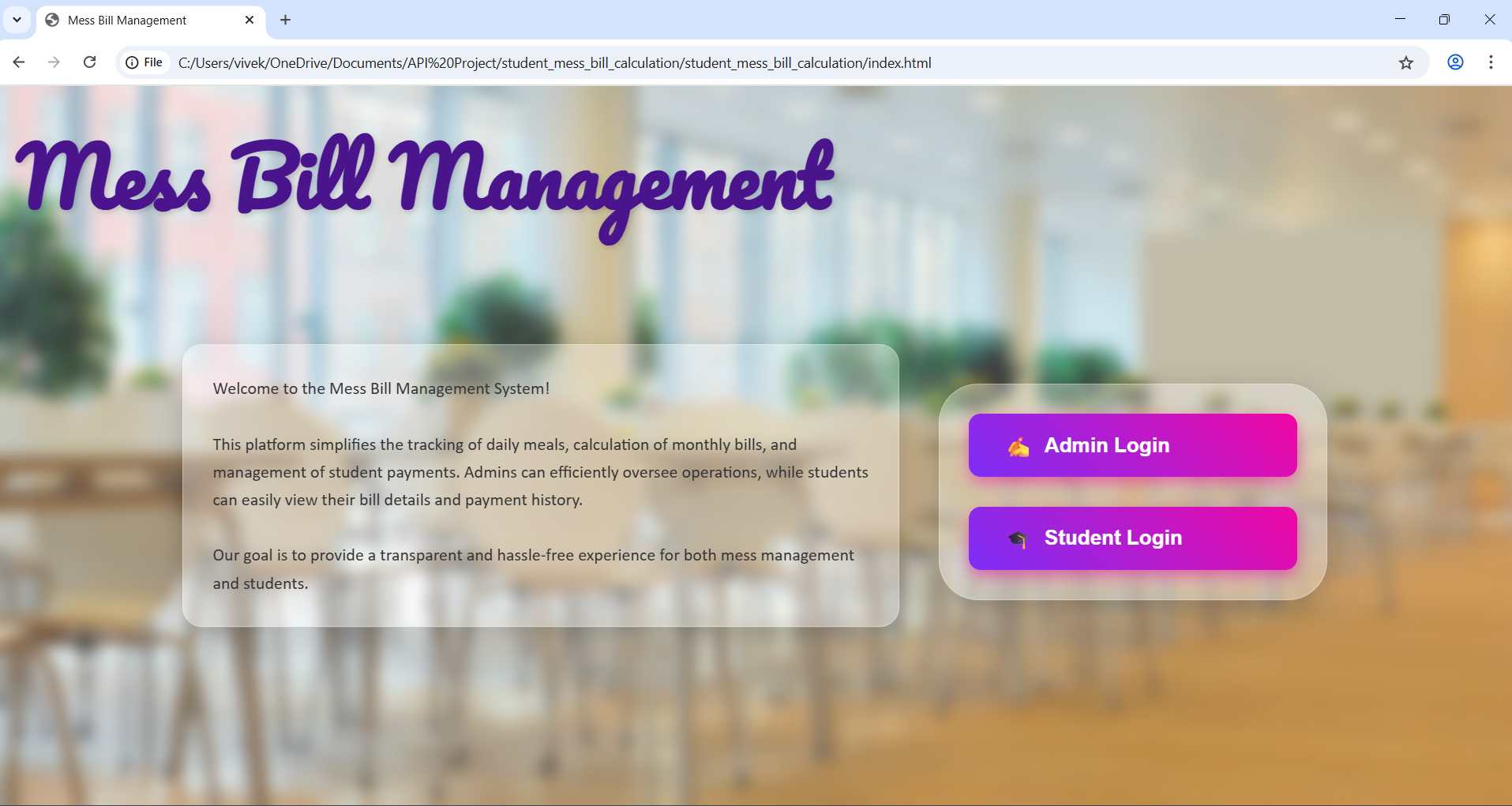
|  |  |
| --- | --- |
| **Table Name** | **Description** |
| student | Student ID, name, branch, phone number |
| admin | Admin credentials |
| expense | Total mess expense for a month |
| attendance | Student attendance per month |
| student\_expense | Monthly calculated student expenses |

# Output

## Screens and Sample Flow

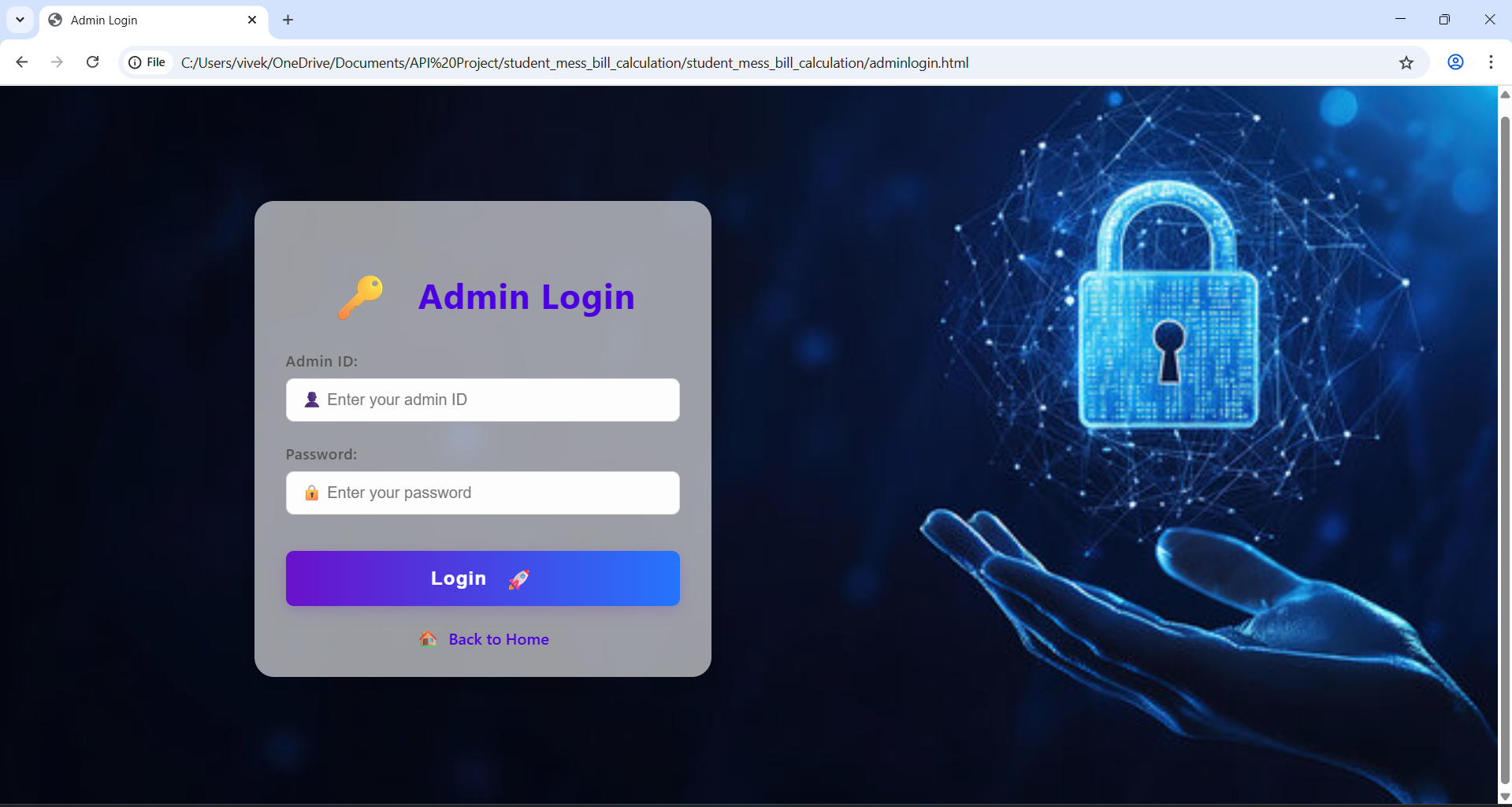
### Home Page (index.html)

* Simple and styled page with buttons:
* Admin Login
* Student Login
* Navigation using Bootstrap grid



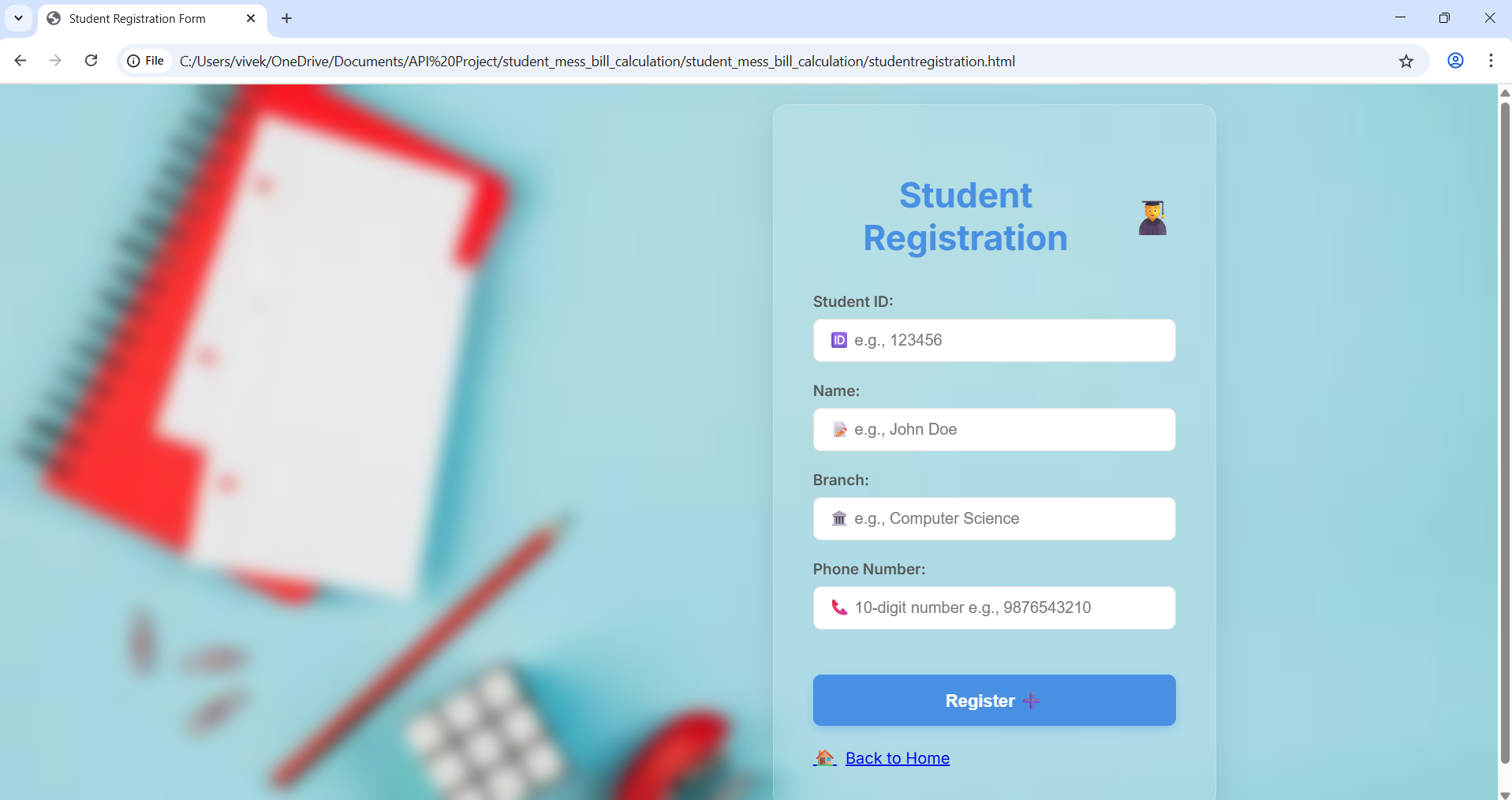
### Admin Login

* Enter Username and Password
* Access to Admin Dashboard
* Display options to manage expenses and attendance



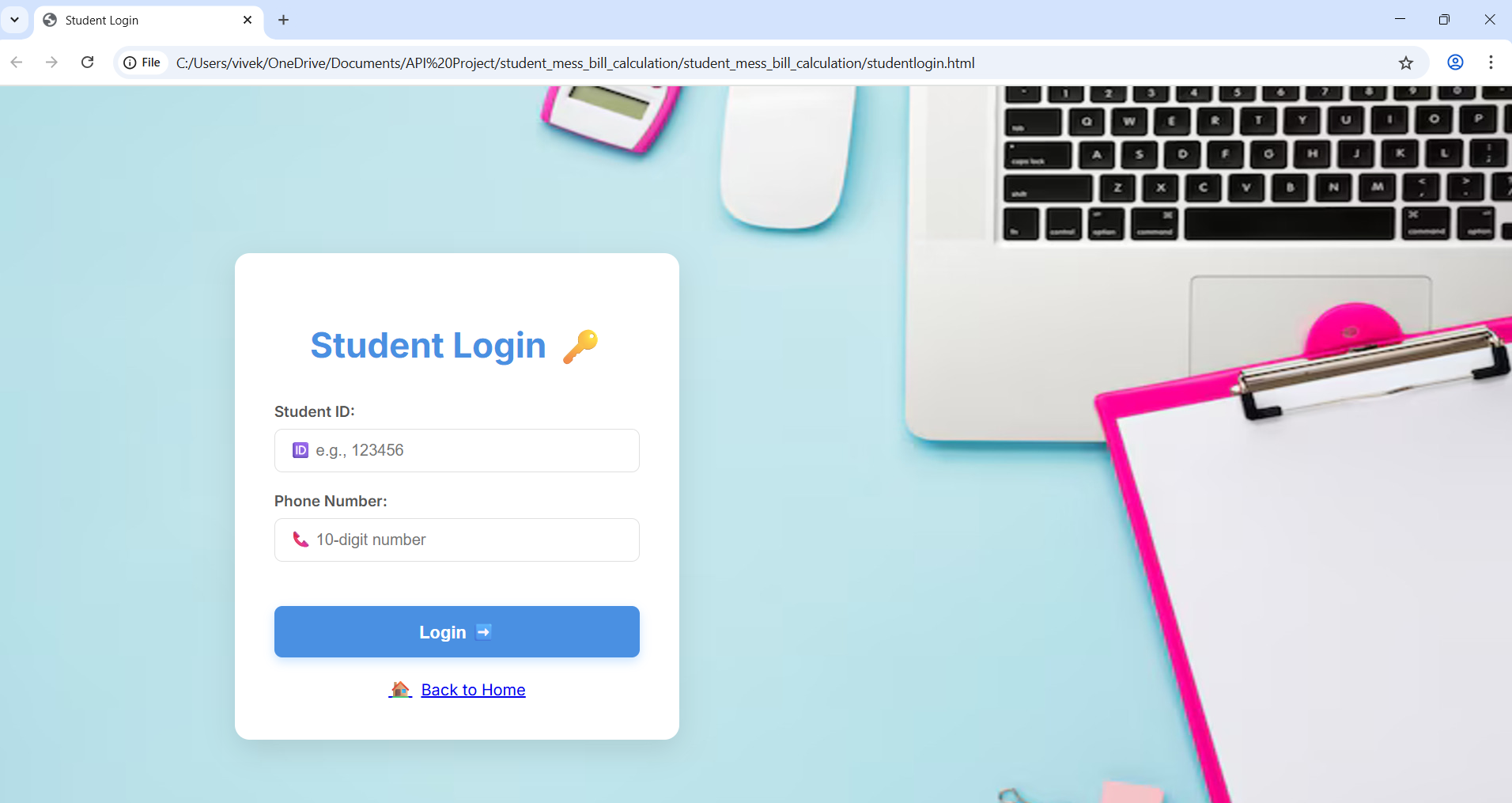
### Student Registration

* Input student details and submit
* Stored in the database
* Confirmation message on success



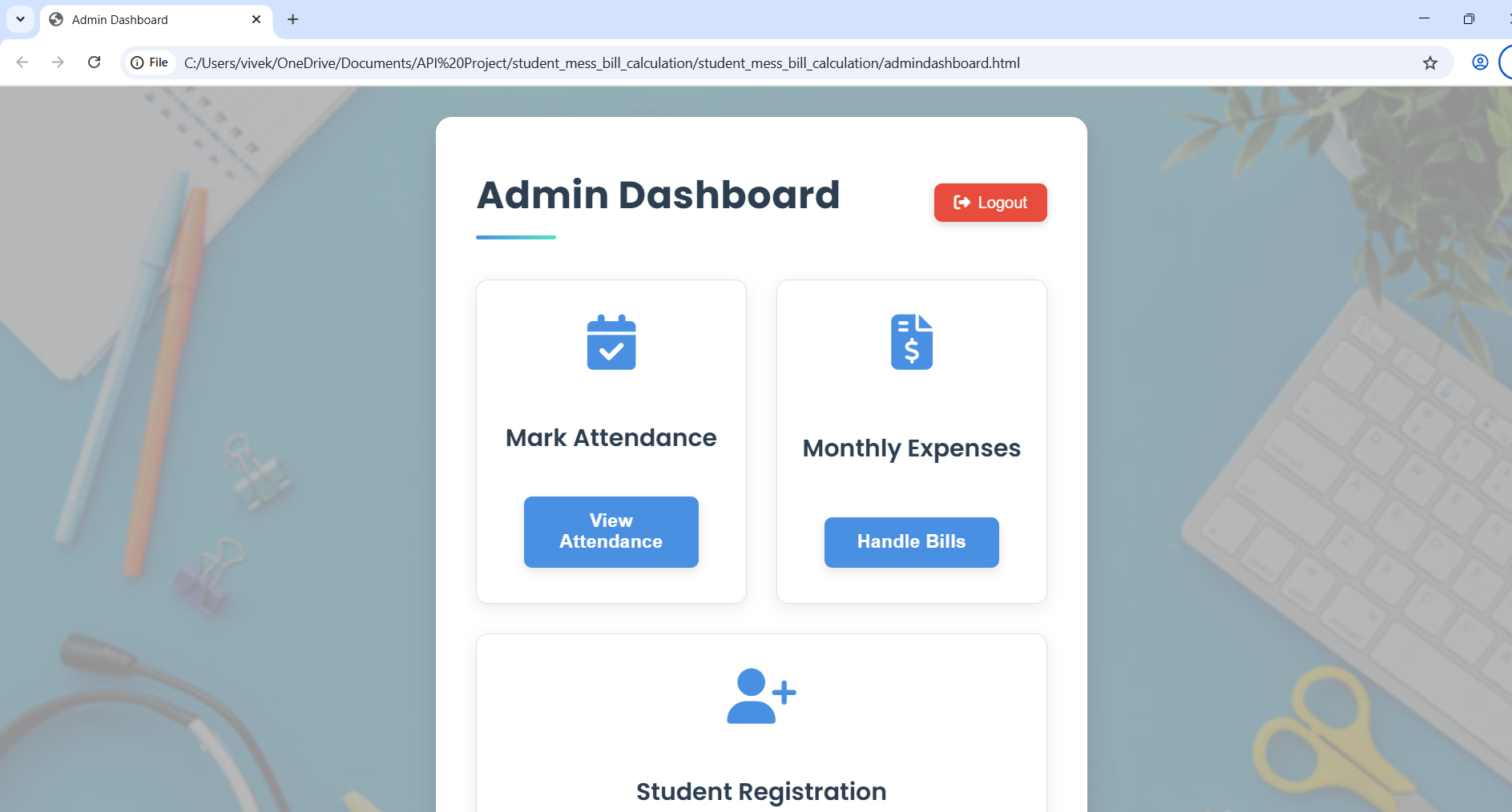
### Student Login

* Student ID and Phone Number required
* Successful login leads to expense dashboard
* Invalid login displays error

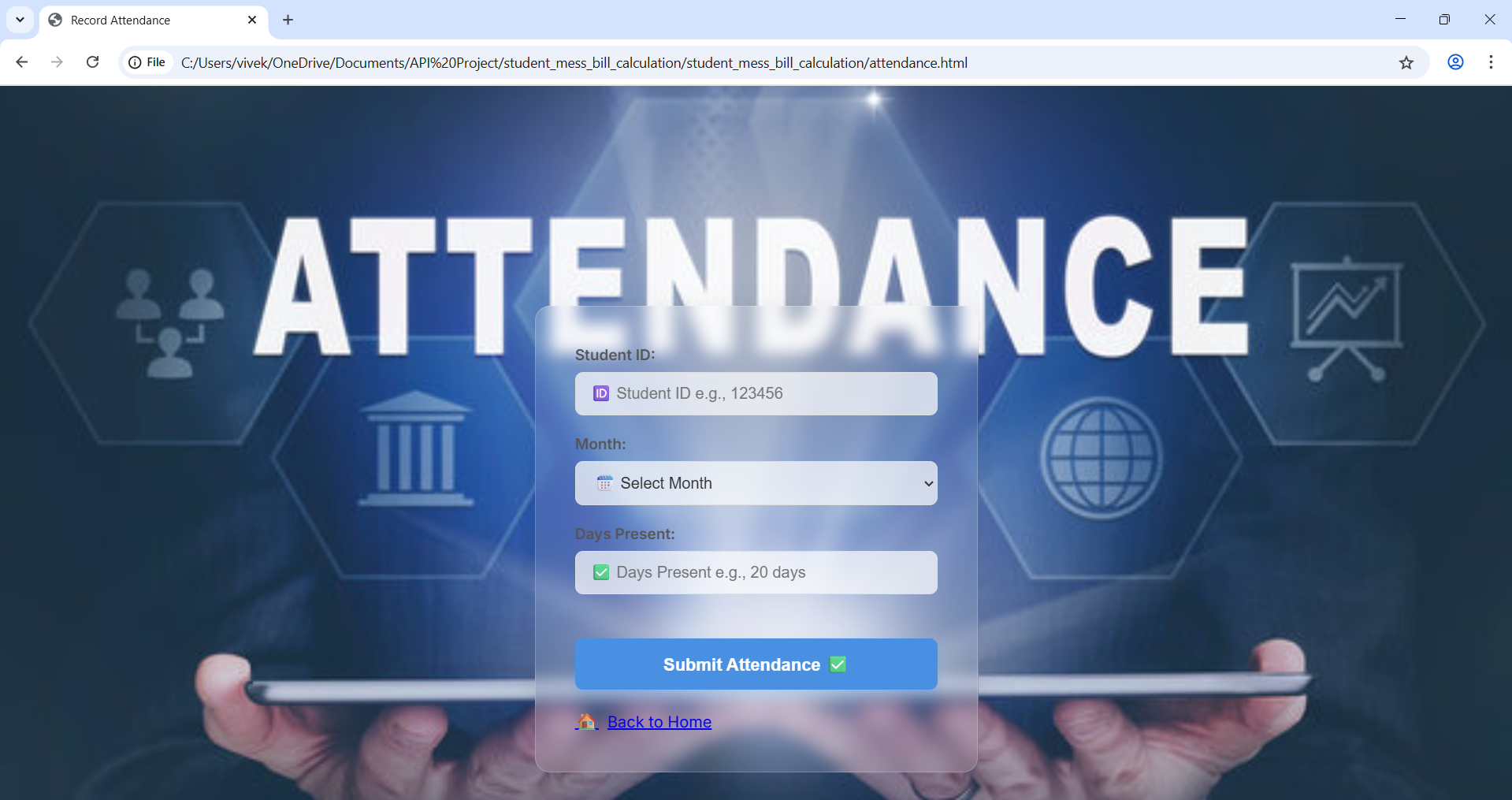


### Admin Dashboard

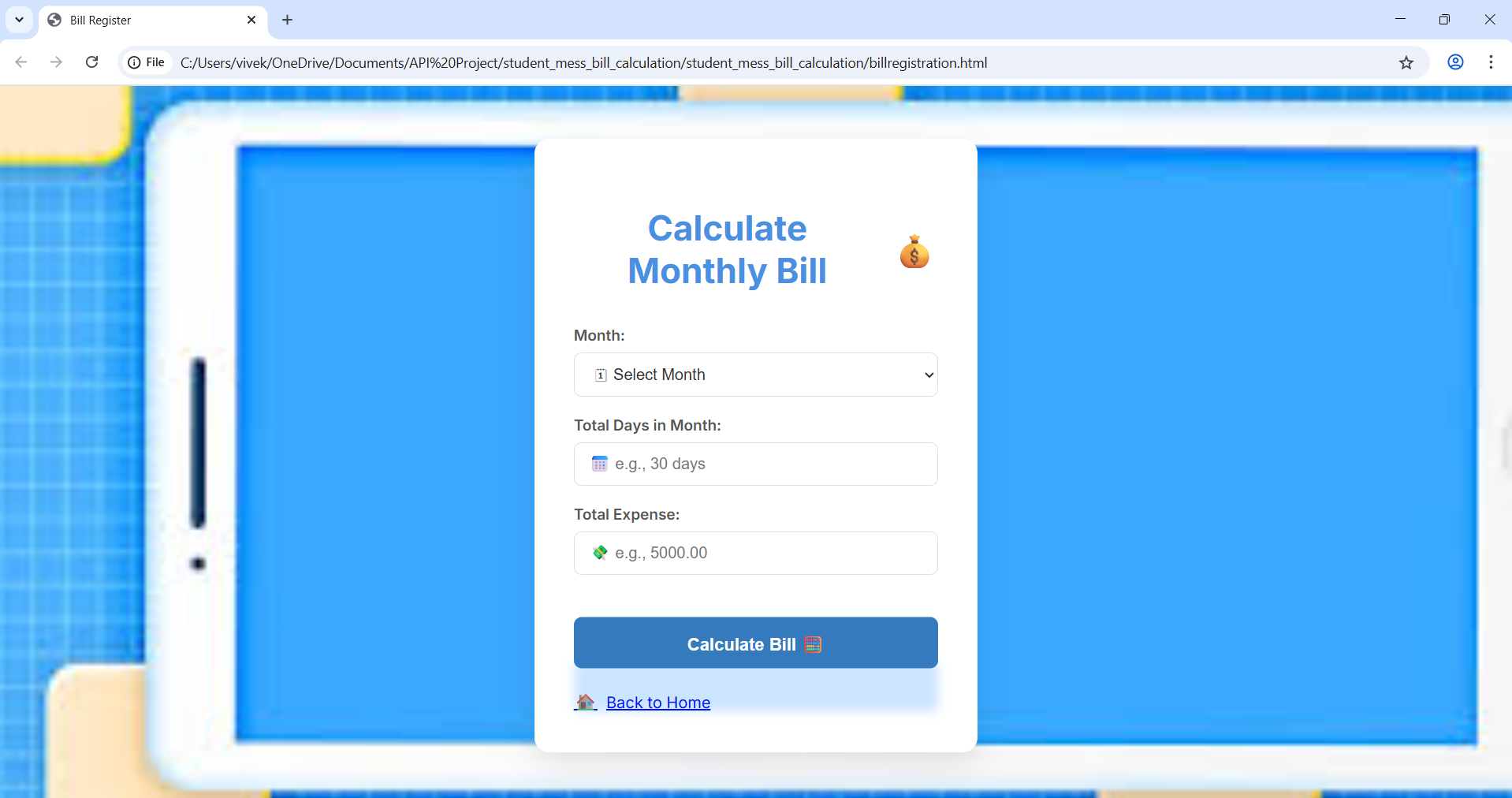
* Displays:
* Monthly Expense
* Attendance Record
* Calculated Bill



# Student Attendance



# Bill Calculation



# Further Research and Enhancements

## Security Improvements

* Encrypt passwords using hashing
* Secure login and session management
* Use HTTPS and input validation techniques
* Implement JWT-based authentication for APIs

## Features to Add

* Admin Dashboard with analytics (charts, tables)
* Attendance marking via calendar interface
* Payment tracking and receipts
* Notification system (email/SMS alerts)
* Mobile-friendly version or app integration

## UI/UX Enhancements

* Use modals for forms and alerts
* Responsive grid layout using CSS Flexbox/Grid
* Pagination and sorting in data tables
* Dark/light theme toggle

## Deployment Plans

* Dockerize app for uniform deployment
* Deploy to Heroku/AWS/GCP with SSL and domain
* Integrate SMTP for email notifications
* Setup CI/CD for smooth updates

## Analytics Module

* Show graphs for:
* Monthly expenses per student
* Attendance trends
* Payment defaulters
* Export reports in Excel/PDF formats

## Code Optimization

* Modularize Flask app (Blueprints)
* Use ORM like SQLAlchemy for DB access
* Write unit tests for major routes

## Scalability

* Support multiple hostels or campuses
* Role-based access: SuperAdmin, Warden, Student
* Load balancing and cloud-based DB for higher throughput