LABORATORY REPORT

Application Development Lab
(CS33002)

B.Tech Program in ECSc

Submitted By

Name: Shreyaa Venkateswaran

Roll No: 2230120

Kalinga Institute of Industrial Technology
(Deemed to be University)
Bhubaneswar, India

Spring 2024-2025

Table of Content

Experiment Number	6
Experiment Title	Database Management Using Flask
Date of Experiment	11-03-2025
Date of Submission	17-03-2025

1. Objective:

To develop an application for user authentication and document sharing.

2. Procedure:

1. Install MySQL workbench in your system and install flask-mysqldb package.

2. Create a database where you wish to store your user name and the password.

3. Implement user authentication/registration form using Flask and the database. For a new user the account is created using the 'signup' button. Existing users can directly login with their credentials.

4. Inside the users can update their personal details, reset their passwords.

5. Inside the users can see the grades for their marks, which they cannot edit personally.

6. Build a responsive frontend for user interactions.

3. Code:

app.py

```
from flask import Flask, render_template, request, redirect, url_for, flash

from flask_mysqldb import MySQL

from flask_bcrypt import Bcrypt
```

```
from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, current_user

app = Flask(__name__)

app.secret_key = "secret123*"


app.config['MYSQL_HOST'] = 'localhost'

app.config['MYSQL_PORT'] = 3306 # Change to 3307 if needed

app.config['MYSQL_USER'] = 'root'

app.config['MYSQL_PASSWORD'] = 'Krishna3*'

app.config['MYSQL_DB'] = 'user_management'

app.config['MYSQL_CURSORCLASS'] = 'DictCursor'

mysql = MySQL(app)

bcrypt = Bcrypt(app)

login_manager = LoginManager(app)

login_manager.login_view = 'login'


class User(UserMixin):

def __init__(self, id, username, email):

self.id = id

self.username = username

self.email = email

@login_manager.user_loader

def load_user(user_id):

conn = mysql.connection

if not conn:
```

```
print("Database connection error!") # Debugging
return None
cur = conn.cursor()
cur.execute("SELECT * FROM users WHERE id = %s", (user_id,))
user = cur.fetchone()
cur.close()
if user:
return User(user["id"], user["username"], user["email"])
return None

@app.route('/')
def home():
return render_template('home.html')

@app.route('/register', methods=['GET', 'POST'])
def register():
if request.method == 'POST':
username = request.form['username']
email = request.form['email']
password = request.form['password']
hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')
cur = mysql.connection.cursor()
try:
cur.execute("INSERT INTO users (username, email, password) VALUES (%s, %s, %s)",
```

```
(username, email, hashed_password))
mysql.connection.commit()
cur.execute("SELECT id FROM users WHERE email = %s", (email,))
user_id = cur.fetchone()
if user_id:
user_id = user_id['id']
else:
flash("Error: User ID not found after insertion!", "danger")
return redirect(url_for('register'))

subjects = ['Mathematics', 'Science', 'English', 'PE', 'Hindi', 'Tamil']
grades = ['A+', 'A', 'A+', 'B', 'B-', 'A+']
for subject, grade in zip(subjects, grades):
cur.execute("INSERT INTO grades (user_id, subject, grade) VALUES (%s, %s, %s)",
(user_id, subject, grade))
mysql.connection.commit()
flash("Registration successful! You can now log in.", "success")
return redirect(url_for('login'))
except Exception as e:
mysql.connection.rollback()
flash("Database error: " + str(e), "danger")
finally:
cur.close()
return render_template('register.html')
```

```
@app.route('/login', methods=['GET', 'POST'])
def login():
if request.method == 'POST':
email = request.form['email']
password = request.form['password']
cur = mysql.connection.cursor()
cur.execute("SELECT * FROM users WHERE email = %s", (email,))
user = cur.fetchone()
cur.close()
if user:
print("Stored Password Hash:", user['password']) # Debugging line
if bcrypt.check_password_hash(user['password'], password):
user_obj = User(user["id"], user["username"], user["email"])
login_user(user_obj)
flash('Login successful!', 'success')
return redirect(url_for('dashboard'))
else:
flash('Invalid email or password.', 'danger')
else:
flash('User not found.', 'danger')
return render_template('login.html')
@app.route('/logout')
@login_required
def logout():
```

```
logout_user()

flash('You have been logged out.', 'info')

return redirect(url_for('login'))


@app.route('/dashboard')

@login_required

def dashboard():

conn = mysql.connection

conn.ping(True)

cur = conn.cursor()

cur.execute("SELECT subject, grade FROM grades WHERE user_id = %s", (current_user.id,))

grades = cur.fetchall()

cur.close()

return render_template('dashboard.html', username=current_user.username, grades=grades)


with app.app_context():

try:

cur = mysql.connection.cursor()

cur.execute("SELECT COUNT(*) FROM users")

count = cur.fetchone()

cur.close()

print(f"Database connection successful! Users count: {count['COUNT(*)']}")

except Exception as e:
```

print("Database Connection Error:", str(e))

if __name__ == '__main__':

app.run(debug=True)

routes.py:

from flask import Blueprint, render_template, request, redirect, url_for, flash, session

from flask_bcrypt import Bcrypt

from flask_login import login_user, logout_user, login_required, current_user

from models import mysql, get_user_by_email

app_routes = Blueprint("app_routes", __name__)

bcrypt = Bcrypt()

@app_routes.route("/")

def home():

return render_template("index.html")

@app_routes.route("/signup", methods=["GET", "POST"])

def signup():

if request.method == "POST":

username = request.form["username"]

email = request.form["email"]

password = bcrypt.generate_password_hash(request.form["password"]).decode("utf-8")

cur = mysql.connection.cursor()

cur.execute("INSERT INTO users (username, email, password) VALUES (%s, %s, %s)", (username, email, password))

```
mysql.connection.commit()
cur.close()
flash("Signup successful! Please log in.", "success")
return redirect(url_for("app_routes.login"))
return render_template("signup.html")
@app_routes.route("/login", methods=["GET", "POST"])
def login():
if request.method == "POST":
email = request.form["email"]
password = request.form["password"]
user = get_user_by_email(email)
if user and bcrypt.check_password_hash(user["password"], password):
login_user(user)
flash("Login successful!", "success")
return redirect(url_for("app_routes.dashboard"))
flash("Invalid credentials. Try again.", "danger")
return render_template("login.html")
@app_routes.route("/dashboard")
@login_required
def dashboard():
return render_template("dashboard.html", username=current_user.username)
@app_routes.route("/logout")
@login_required
def logout():
```

```
logout_user()
flash("You have been logged out.", "success")
return redirect(url_for("app_routes.login"))
```

db_init.py:

```
CREATE DATABASE user_management;
USE user_management;
CREATE TABLE users (
id INT AUTO_INCREMENT PRIMARY KEY,
username VARCHAR(100) NOT NULL UNIQUE,
email VARCHAR(100) NOT NULL UNIQUE,
password VARCHAR(255) NOT NULL
);
CREATE TABLE grades (
id INT AUTO_INCREMENT PRIMARY KEY,
user_id INT,
subject VARCHAR(100),
grade VARCHAR(10),
FOREIGN KEY (user_id) REFERENCES users(id)
);
```

models.py:

```
from flask_mysqldb import MySQL
mysql = MySQL()
def get_user_by_email(email):
```

```
cur = mysql.connection.cursor()

cur.execute("SELECT * FROM users WHERE email = %s", (email,))

user = cur.fetchone()

cur.close()

return user
```

base.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>{% block title %}Flask App{% endblock %}</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container">

<a class="navbar-brand" href="{{ url_for('home') }}">Flask App</a>

<div class="d-flex">

{% if current_user.is_authenticated %}
```

Dashboard

Logout

{% else %}

Login

Register

{% endif %}

</nav>

{% with messages = get_flashed_messages(with_categories=True) %}

{% if messages %}

{% for category, message in messages %}

{{ message }}

<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

{% endfor %}

{% endif %}

{% endwith %}

{% block content %}{% endblock %}

```
</body>
</html>
```

dashboard.html:

```
{% extends "base.html" %}
{% block title %}Dashboard{% endblock %}
{% block content %}
<div class="container mt-4">
<h2 class="text-center">Welcome, {{ username }}!</h2>
<div class="card shadow-lg mt-3">
<div class="card-header bg-success text-white text-center">
<h4>Your Grades</h4>
</div>
<div class="card-body">
<table class="table table-striped text-center">
<thead class="table-dark">
<tr>
<th>Subject</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
{% for grade in grades %}
<tr>
```

```
<td>{{ grade.subject }}</td>
<td>{{ grade.grade }}</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>
{% endblock %}
```

grades.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Your Grades</title>
</head>
<body>
<h2>Your Grades</h2>
<table border="1">
<tr>
<th>Subject</th>
<th>Grade</th>
```

```
</tr>
{% for grade in grades %}
<tr>
<td>{{ grade.subject }}</td>
<td>{{ grade.grade }}</td>
</tr>
{% endfor %}
</table>
<a href="/">Back to Home</a>
</body>
</html>
```

home.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Home - User Management System</title>
<style>
body {
background-color: #f4f4f4;
font-family: Arial, sans-serif;
text-align: center;
margin: 0;
```

```
padding: 0;
}
.container {
width: 90%;
max-width: 500px;
margin: 100px auto;
background: white;
padding: 20px;
border-radius: 10px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}
h1 {
color: #333;
}
p {
color: #666;
font-size: 16px;
}
.btn {
display: inline-block;
padding: 10px 20px;
margin: 10px;
text-decoration: none;
color: white;
background-color: #4CAF50;
```

```
border-radius: 5px;

transition: 0.3s;

}

.btn:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div class="container">

<h1>Welcome!</h1>

<p>Please login to view your grades. Kindly register if not done already.</p>

<a href="{{ url_for('register') }}" class="btn">Register</a>

<a href="{{ url_for('login') }}" class="btn">Login</a>

</div>

</body>

</html>
```

index.html:

```
{% extends "base.html" %}

{% block content %}

<h2>Welcome</h2>

<a href="{{ url_for('app_routes.signup') }}">Sign Up</a> |

<a href="{{ url_for('app_routes.login') }}">Login</a>
```

{% endblock %}

login.html:

```
{% extends "base.html" %}
{% block title %}Login{% endblock %}
{% block content %}
<div class="container mt-5">
<div class="row justify-content-center">
<div class="col-md-5">
<div class="card shadow-lg">
<div class="card-header bg-dark text-white text-center">
<h4>Login</h4>
</div>
<div class="card-body">
<form method="POST">
<div class="mb-3">
<label class="form-label">Email</label>
<input type="email" class="form-control" name="email" required>
</div>
<div class="mb-3">
<label class="form-label">Password</label>
<input type="password" class="form-control" name="password" required>
</div>
<button type="submit" class="btn btn-dark w-100">Login</button>
</form>
```

```
</div>
<div class="card-footer text-center">
Don't have an account? <a href="{{ url_for('register') }}">Register</a>
</div>
</div>
</div>
</div>
</div>
{% endblock %}
```

register.html:

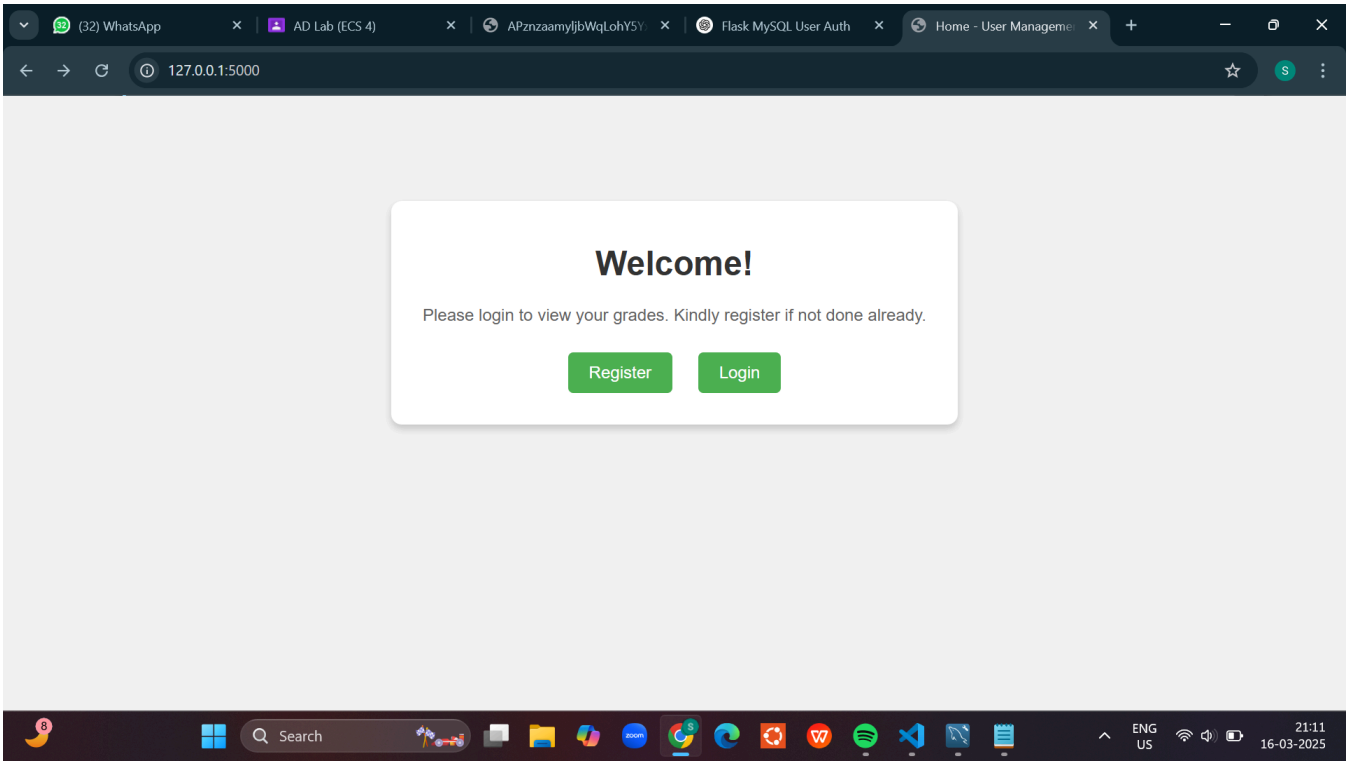
```
{% extends "base.html" %}
{% block title %}Register{% endblock %}
{% block content %}
<div class="container mt-5">
<div class="row justify-content-center">
<div class="col-md-6">
<div class="card shadow-lg">
<div class="card-header bg-primary text-white text-center">
<h4>Create an Account</h4>
</div>
<div class="card-body">
<form method="POST">
<div class="mb-3">
<label class="form-label">Username</label>
```

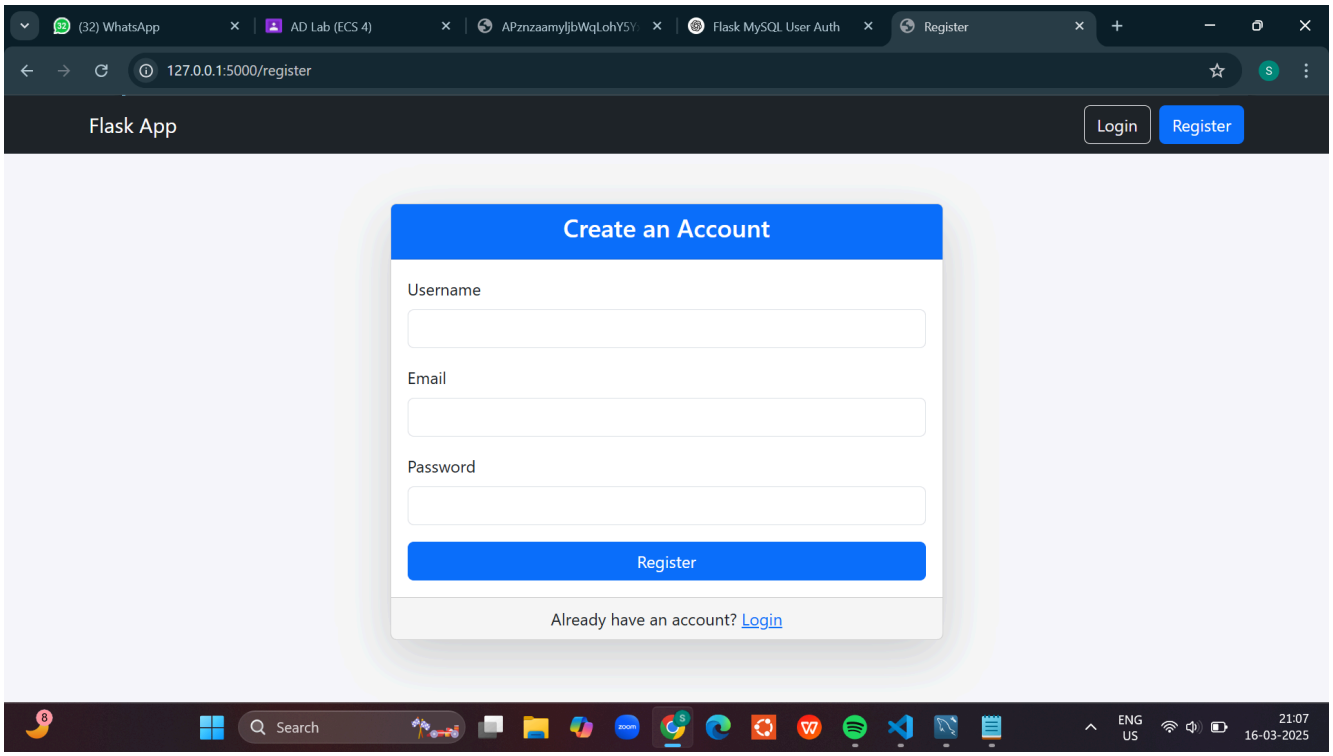
```
<input type="text" class="form-control" name="username" required>

</div>

<div class="mb-3">

<label class="form-label">Email</label>

<input type="email" class="form-control" name="email" required>

</div>

<div class="mb-3">

<label class="form-label">Password</label>

<input type="password" class="form-control" name="password" required>

</div>

<button type="submit" class="btn btn-primary w-100">Register</button>

</form>

</div>

<div class="card-footer text-center">

Already have an account? <a href="{{ url_for('login') }}">Login</a>

</div>

</div>

</div>

</div>

</div>

{% endblock %}
```

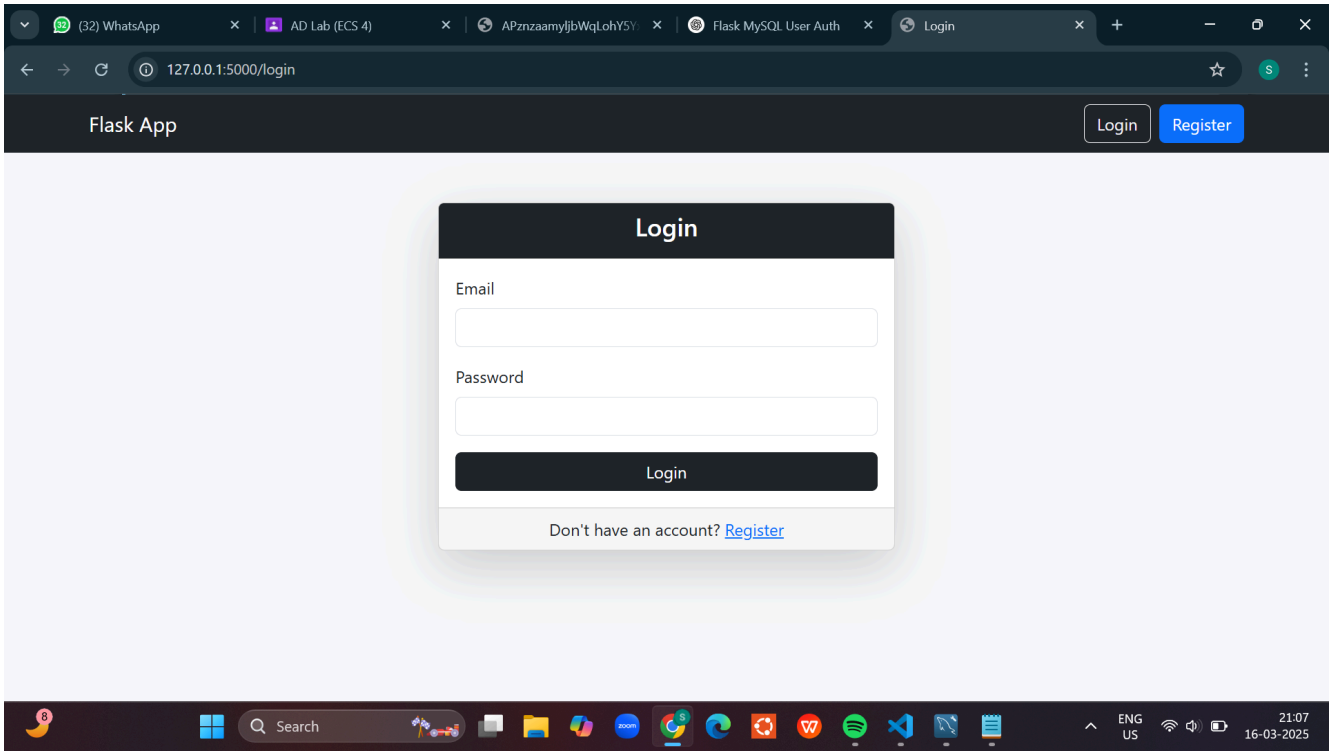
signup.html:

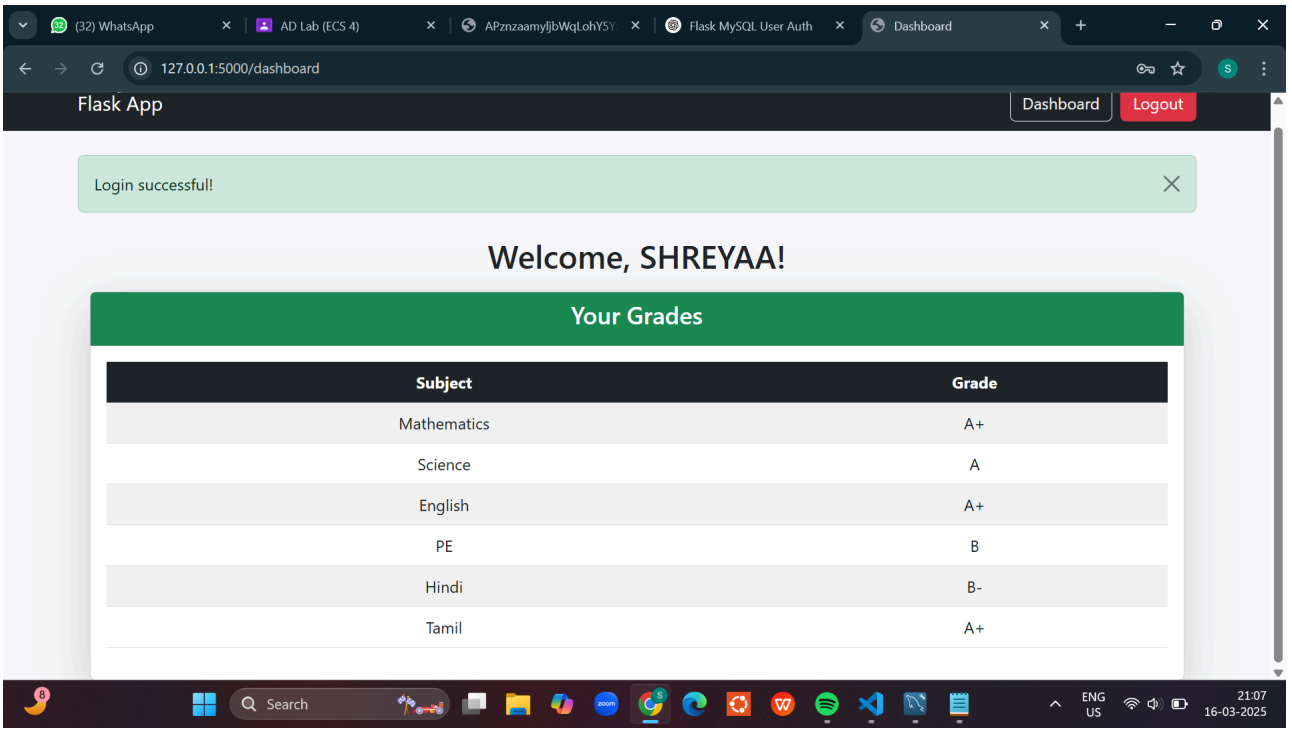
```
{% extends "base.html" %}

{% block content %}

<form method="POST">

<input type="text" name="username" placeholder="Username" required>

<input type="email" name="email" placeholder="Email" required>

<input type="password" name="password" placeholder="Password" required>

<button type="submit">Sign Up</button>

</form>

{% endblock %}
```

4. Results/Output:




(32) WhatsApp
AD Lab (ECS 4)
Flask MySQL User Auth
Register
127.0.0.1:5000/register
Flask App
Login
Register
Create an Account
Username
Email
Password
Register
Already have an account? Login
Search
ENG
US
21:07
16-03-2025


(32) WhatsApp
AD Lab (ECS 4)
Flask MySQL User Auth
Login
127.0.0.1:5000/login
Flask App
Login
Register
Login
Email
Password
Login
Don't have an account? Register
Search
ENG
US
21:07
16-03-2025



5. Remarks:

Created a user dashboard that allows users to register or login. The user credentials are then added to the user table in the database that contains the grades of the user as well. The flask application is linked to the sql workbench (database: user_management). The user_id is automatically assigned and the passwords provided by the user are hashed and stored in the database. Accordingly, the grades are retrieved for each user and displayed in the web page.

Website link: DB_Management

GitHub link: GitHub

Shreyaa Venkateswaran

Signature of the Lab Coordinator