

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

Exp No.	Title	Date of Experiment	Date of Submission	Remarks
1.	Experiment 1: Build a resume using HTML/CSS	07-01-2025	14-01-2025	
2.	Experiment 2: Machine Learning for Cat and Dog Classification	15-01-2025	20-01-2025	
3.	Experiment 3: Regression Analysis for Stock Prediction	21-01-2025	27-01-2025	
4.	Experiment 4: Conversational Chatbot with Any Files	04-02-2025	09-02-2025	
5.	Experiment 5: Web Scraper using LLMs	16-02-2025	17-03-2025	
6.	Experiment 6: Database Management Using Flask	11-03-2025	17-03-2025	
7.				
8.				
9.	Open Ended 1			
10.	Open Ended 2			

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:

auth.py:

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

```
from models.user import User, Grade
```

```
from extensions import db

auth_bp = Blueprint('auth', __name__)

@auth_bp.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        user = User.query.filter_by(username=username).first()
        if user and user.check_password(password):
            session['user_id'] = user.id
            flash('Login successful!', 'success')
            return redirect(url_for('dashboard.dashboard'))
        else:
            flash('Invalid username or password', 'error')
            return render_template('login.html')

@auth_bp.route('/signup', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        # Check if the user already exists
        user = User.query.filter_by(username=username).first()
        if user:
            flash('Username already exists', 'error')
```

```
else:

# Create a new user

new_user = User(username=username, email=email)

new_user.set_password(password)

db.session.add(new_user)

db.session.commit()

# Add sample grades for the new user

sample_grades = [

Grade(user_id=new_user.id, subject='Math', grade='A+'),

Grade(user_id=new_user.id, subject='Science', grade='A'),

Grade(user_id=new_user.id, subject='History', grade='A'),

Grade(user_id=new_user.id, subject='Hindi', grade='B'),

Grade(user_id=new_user.id, subject='English', grade='A+'),

Grade(user_id=new_user.id, subject='PE', grade='B'),

Grade(user_id=new_user.id, subject='Tamil', grade='A+'),

]

db.session.add_all(sample_grades)

db.session.commit()

# Log the user in after registration

session['user_id'] = new_user.id

flash('Account created and login successful!', 'success')

return redirect(url_for('dashboard.dashboard'))

return render_template('signup.html')

@auth_bp.route('/logout')

def logout():
```

```
session.pop('user_id', None)

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

return redirect(url_for('auth.login'))
```

dashboard.py:

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

from models.user import User

from extensions import db

dashboard_bp = Blueprint('dashboard', __name__)

@dashboard_bp.route('/dashboard')
def dashboard():
    user_id = session.get('user_id')

    if user_id:
        user = User.query.get(user_id)

        return render_template('dashboard.html', user=user)

    return "User not logged in", 404

@dashboard_bp.route('/update_profile', methods=['GET', 'POST'])
def update_profile():
    user_id = session.get('user_id')

    if not user_id:
        return "User not logged in", 404

    user = User.query.get(user_id)

    if request.method == 'POST':
        new_username = request.form.get('username')
```

```
new_email = request.form.get('email')

if new_username:
    user.username = new_username

if new_email:
    user.email = new_email

db.session.commit()

flash('Profile updated successfully!', 'success')

return redirect(url_for('dashboard.dashboard'))

return render_template('update_profile.html', user=user)
```

grades.py:

```
from flask import Blueprint, render_template, session
from models.user import Grade

grades_bp = Blueprint('grades', __name__)

@grades_bp.route('/grades')
def grades():
    user_id = session.get('user_id')

    if user_id:
        grades = Grade.query.filter_by(user_id=user_id).all()
        return render_template('grades.html', grades=grades)

    return "User not logged in", 404
```

styles.css:

```
body {

font-family: 'Arial', sans-serif;
```

```
margin: 0;

padding: 0;

background-color: #f4f4f9;

color: #333;

}

header {

background-color: #6200ea;

color: white;

padding: 1rem 2rem;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

nav {

display: flex;

justify-content: space-between;

align-items: center;

}

.logo a {

font-size: 1.5rem;

font-weight: bold;

color: white;

text-decoration: none;

}

.nav-links {

list-style: none;

display: flex;
```



```
gap: 1.5rem;
margin: 0;
padding: 0;
}

.nav-links li a {
color: white;
text-decoration: none;
font-weight: 500;
transition: color 0.3s ease;
}

.nav-links li a:hover {
color: #bb86fc;
}

main {
padding: 2rem;
}

footer {
text-align: center;
padding: 1rem;
background-color: #333;
color: white;
margin-top: 2rem;
}

/* Flash Messages */

.flash-message {
```

```
padding: 1rem;
margin-bottom: 1rem;
border-radius: 4px;
text-align: center;
}

.flash-message.success {
background-color: #c8e6c9;
color: #2e7d32;
}

.flash-message.error {
background-color: #ffcdd2;
color: #c62828;
}

/* Forms */

form {
max-width: 400px;
margin: 0 auto;
padding: 2rem;
background: white;
border-radius: 8px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

form label {
display: block;
margin-bottom: 0.5rem;
```

```
font-weight: 500;
}

form input {
width: 100%;
padding: 0.75rem;
margin-bottom: 1rem;
border: 1px solid #ddd;
border-radius: 4px;
font-size: 1rem;
}

form button {
width: 100%;
padding: 0.75rem;
background-color: #6200ea;
color: white;
border: none;
border-radius: 4px;
font-size: 1rem;
cursor: pointer;
transition: background-color 0.3s ease;
}

form button:hover {
background-color: #3700b3;
}

/* Tables */
```

```
table {  
width: 100%;  
border-collapse: collapse;  
margin-top: 1rem;  
background: white;  
border-radius: 8px;  
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
}  
table th, table td {  
padding: 1rem;  
text-align: left;  
border-bottom: 1px solid #ddd;  
}  
table th {  
background-color: #6200ea;  
color: white;  
}  
table tr:hover {  
background-color: #f1f1f1;  
}  
/* Buttons */  
.btn {  
display: inline-block;  
padding: 0.75rem 1.5rem;  
background-color: #6200ea;
```

```
color: white;

text-decoration: none;

border-radius: 4px;

transition: background-color 0.3s ease;

}

.btn:hover {

background-color: #3700b3;

}

/* Utility Classes */

.text-center {

text-align: center;

}

.mt-4 {

margin-top: 2rem;

}
```

base.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Flask App</title>

<!-- Custom CSS -->
```

```

<link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">

</head>

<body>

<header>

<nav>

<div class="logo">

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

</div>

<ul class="nav-links">

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

<li><a href="{{ url_for('auth.signup') }}">Signup</a></li>

<li><a href="{{ url_for('dashboard.dashboard') }}">Dashboard</a></li>

<li><a href="{{ url_for('grades.grades') }}">Grades</a></li>

</ul>

</nav>

</header>

<main>

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

{% if messages %}

{% for category, message in messages %}

<div class="flash-message {{ category }}">

{{ message }}

</div>

{% endfor %}

{% endif %}

```

```
{% endwith %}

{% block content %} {% endblock %}

</main>

</body>

</html>
```

dashboard.html:

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

{% block content %}

<div class="text-center">

<h1>Dashboard</h1>

<p>Welcome, {{ user.username }}!</p>

<a href="{{ url_for('dashboard.update_profile') }}" class="btn">Update
Profile</a>

</div>

{% endblock %}
```

grades.html:

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

{% block content %}

<div class="text-center">

<h1>Your Grades</h1>

<table>

<thead>

<tr>
```

```

<th>Subject</th>

<th>Grade</th>

</tr>

</thead>

<tbody>

{% for grade in grades %}

<tr>

<td>{{ grade.subject }}</td>

<td>{{ grade.grade }}</td>

</tr>

{% else %}

<tr>

<td colspan="2">No grades found.</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

{% endblock %}

```

index.html:

```

{% extends "base.html" %}

{% block content %}

<div class="text-center mt-4">

<h1>Welcome to the Flask App!</h1>

```


<p>This is a simple Flask application for user authentication and document sharing.</p>

<div class="mt-4">

Login

Signup

</div>

</div>

{% endblock %}

login.html:

{% extends "base.html" %}

{% block content %}

<div class="text-center">

<h1>Login</h1>

<form method="POST">

<label for="username">Username</label>

<input type="text" id="username" name="username" required>

<label for="password">Password</label>

<input type="password" id="password" name="password" required>

<button type="submit">Login</button>

</form>

<p class="mt-4">Don't have an account? Sign up</p>

</div>

{% endblock %}

signup.html:

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

{% block content %}

<div class="text-center">

<h1>Signup</h1>

<form method="POST">

<label for="username">Username</label>

<input type="text" id="username" name="username" required>

<label for="email">Email</label>

<input type="email" id="email" name="email" required>

<label for="password">Password</label>

<input type="password" id="password" name="password" required>

<button type="submit">Signup</button>

</form>

<p class="mt-4">Already have an account? <a href="{{
url_for('auth.login') }}">Login</a></p>

</div>

{% endblock %}
```

update_profile.html:

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

{% block content %}

<div class="text-center">

<h1>Update Profile</h1>

<form method="POST">
```

```

<label for="username">Username</label>

<input type="text" id="username" name="username" value="{{
user.username }}">

<label for="email">Email</label>

<input type="email" id="email" name="email" value="{{ user.email
}}">

<button type="submit">Update Profile</button>

</form>

<p class="mt-4"><a href="{{ url_for('dashboard.dashboard') }}">Back to
Dashboard</a></p>

</div>

{% endblock %}

```

app.py:

```

from flask import Flask, render_template

from config import Config

from extensions import db

from routes.auth import auth_bp

from routes.dashboard import dashboard_bp

from routes.grades import grades_bp

app = Flask(__name__)

app.config.from_object(Config)

# Initialize the database

db.init_app(app)

# Register blueprints

app.register_blueprint(auth_bp)

```

```
app.register_blueprint/dashboard_bp)

app.register_blueprint(grades_bp)

@app.route('/')

def home():

return render_template('index.html')

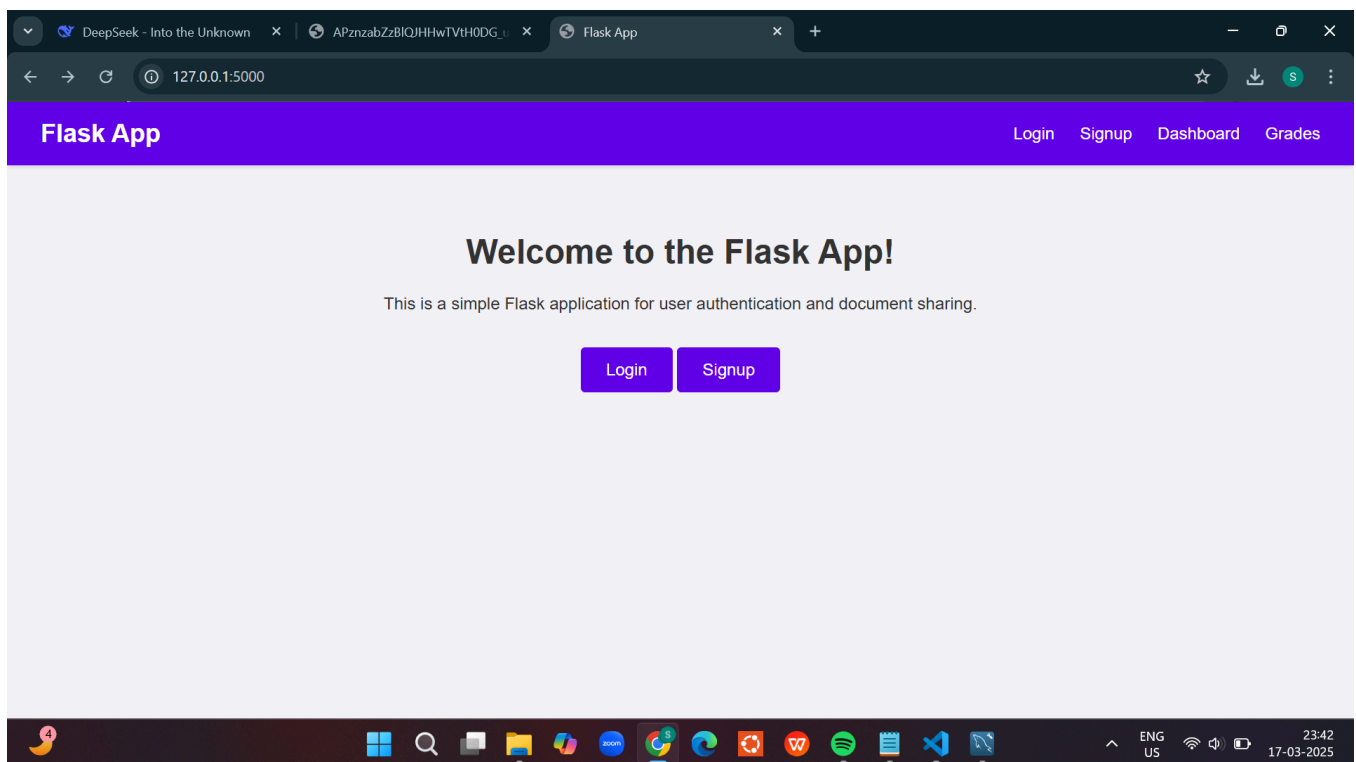
if __name__ == '__main__':

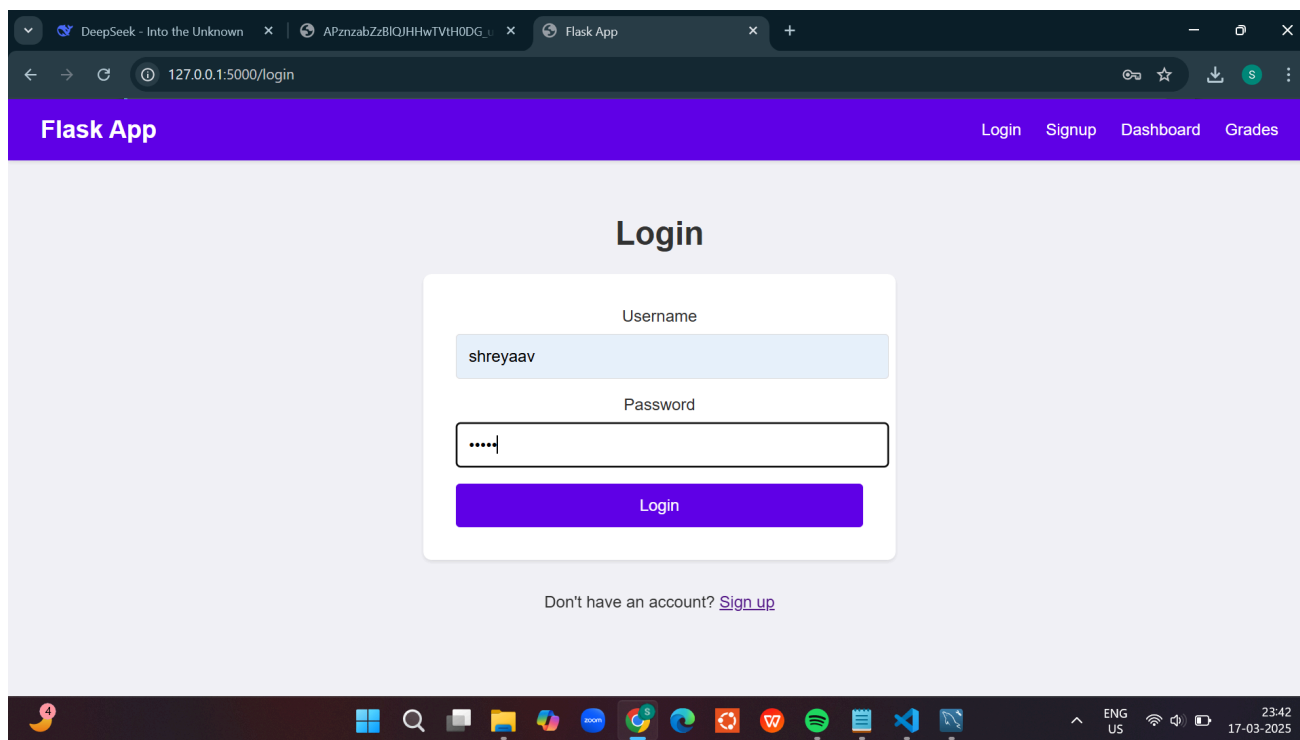
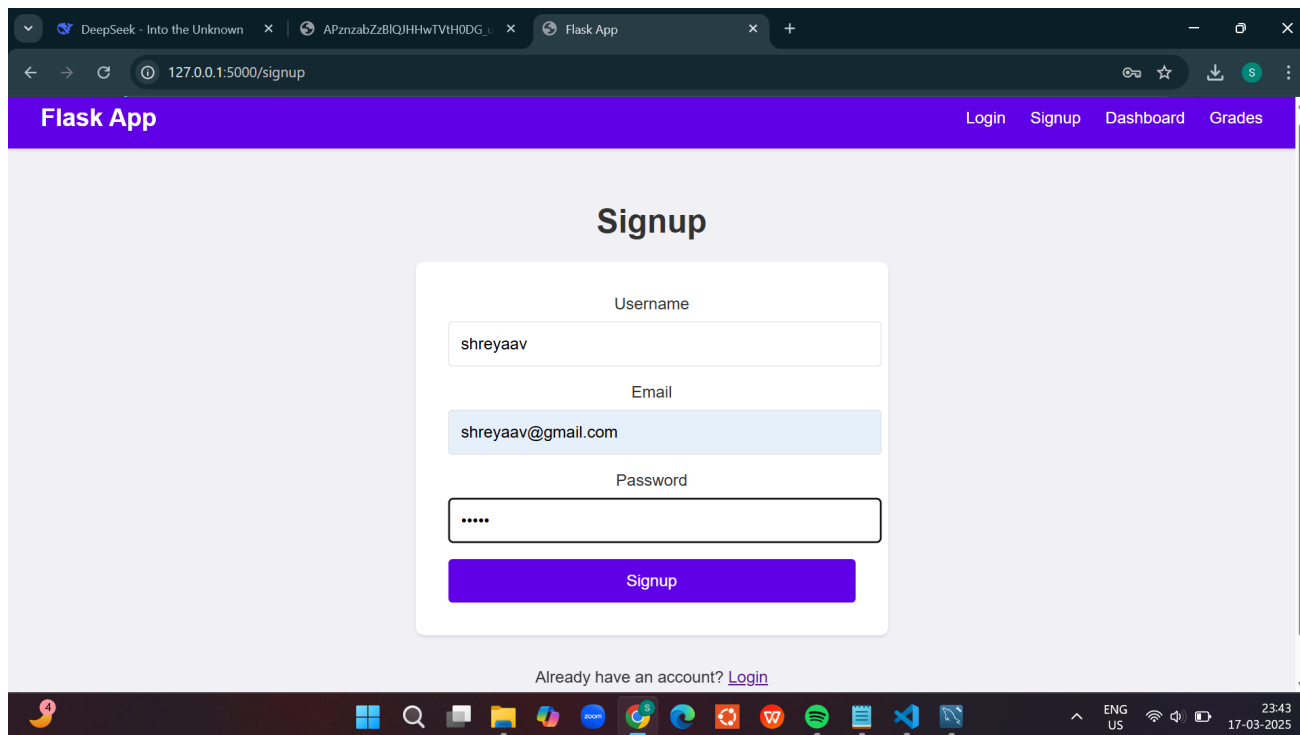
with app.app_context():

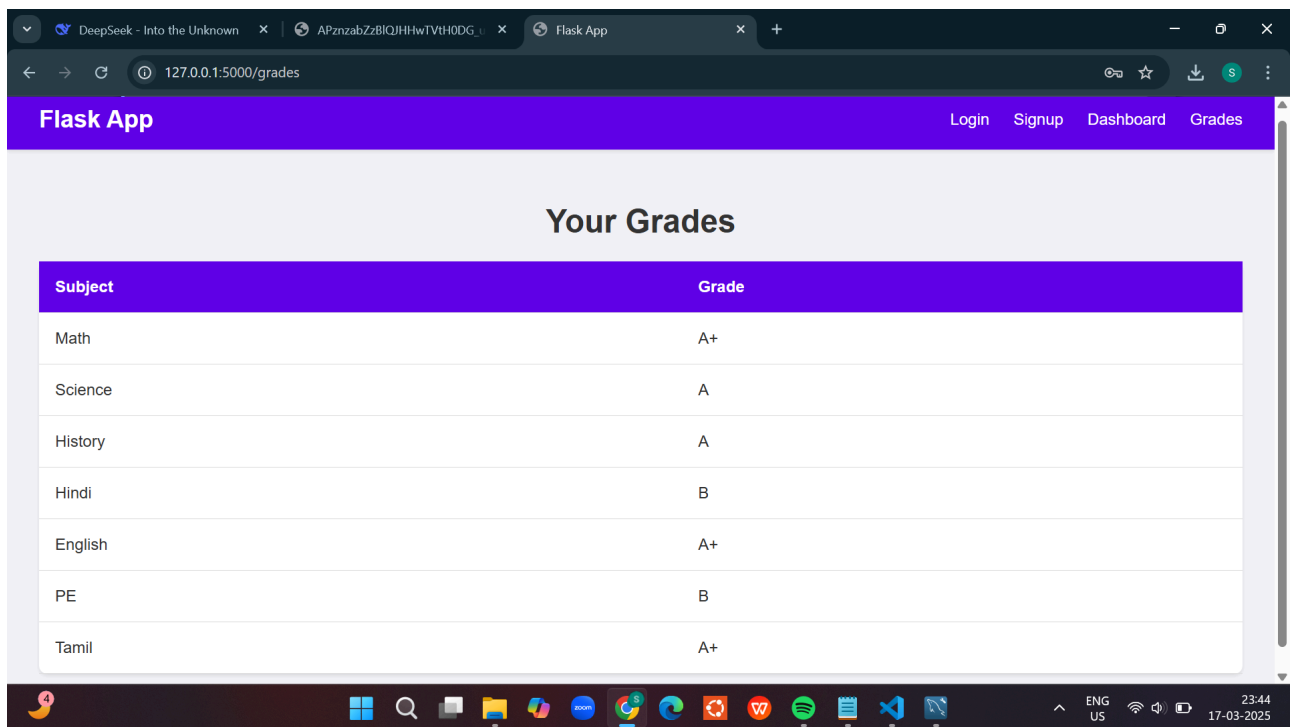
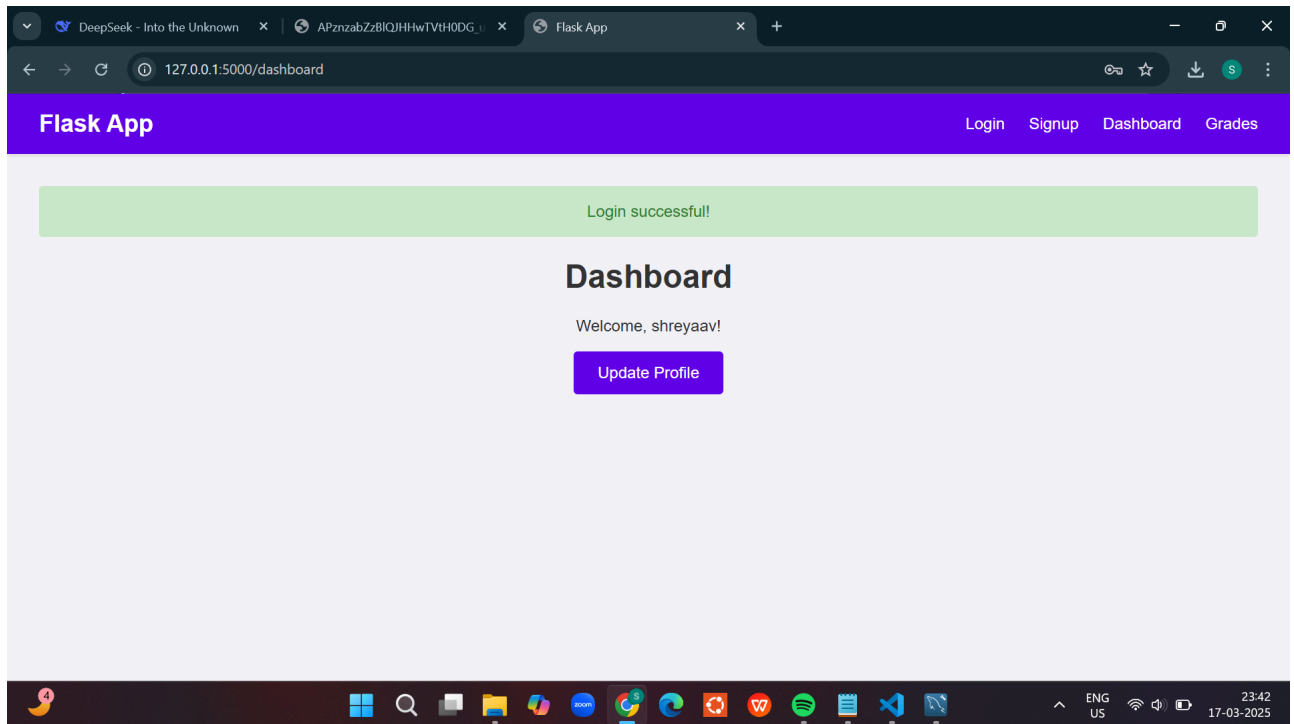
db.create_all() # Create database tables

app.run(debug=True)
```

4. Results/Output:







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
