LABORATORY REPORT

Application Development Lab (CS33002)

B.Tech Program in ECSc

Submitted By

Name:-Pruthibiraj Nayak

Roll No: 2230183

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:- (Steps Followed)

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.
4. For a new user the account is created using the 'signup' button. Existing users can directly login with their crendtials.
5. Inside the users can update their personal details, reset their passwords.
6. Inside the users can see the grades for their marks, which they cannot edit personally
7. Build a responsive frontend for user interactions.

Code:-

FLASK CODE

```
from flask import Flask, render_template, request, redirect, url_for, session, flash
from flask_login import LoginManager, UserMixin, login_user, login_required, logout_user, current_user
from flask_bcrypt import Bcrypt
from db import mysql, init_db
import MySQLdb.cursors
import random

app = Flask(__name__)
bcrypt = Bcrypt(app)
init_db(app)
# Flask-Login setup
login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = "login"

class User(UserMixin):
    def __init__(self, id, username, email, details=""):
        self.id = id
        self.username = username
        self.email = email
        self.details = details
```

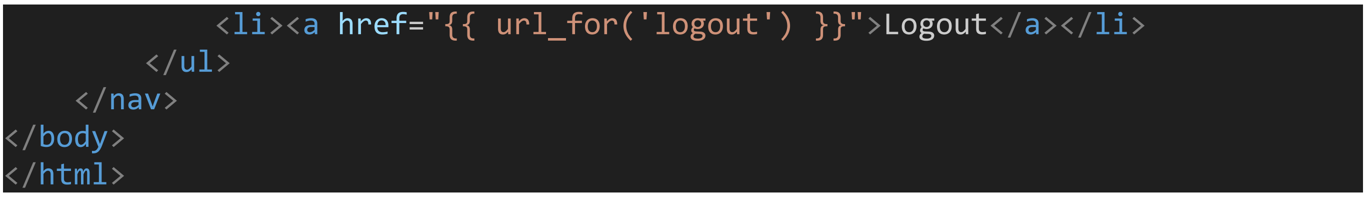
```
@login_manager.user_loader
def load_user(user_id):
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("SELECT * FROM users WHERE id=%s", (user_id,))
    user = cursor.fetchone()
    cursor.close()
    if user:
        return User(user["id"], user["username"], user["email"],
user.get("details", ""))
    return None

# Home Page
@app.route("/")
def home():
    return render_template("index.html")

# Signup
@app.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")
        cursor = mysql.connection.cursor()
        cursor.execute("INSERT INTO users (username, email, password) VALUES
(%s, %s, %s)", (username, email, password))
        mysql.connection.commit()
        cursor.close()
        flash("Signup successful! Please login.", "success")
        return redirect(url_for("login"))
    return render_template("signup.html")

# Login
@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form["email"]
        password = request.form["password"]
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM users WHERE email=%s", [email])
        user = cursor.fetchone()
        cursor.close()
        if user and bcrypt.check_password_hash(user["password"], password):
            user_obj = User(user["id"], user["username"], user["email"])
            login_user(user_obj)
            return redirect(url_for("dashboard"))
        else:
            flash("Invalid credentials. Please try again.", "danger")
    return render_template("login.html")

# Dashboard
@app.route("/dashboard")
@login_required
def dashboard():
    marks = random.randint(50, 100)
    grade = 'A' if marks >= 85 else 'B' if marks >= 70 else 'C'
    return render_template("dashboard.html",
username=current_user.username,marks=marks, grade=grade)

# Update Personal Details (Username, Email, Additional Details)
@app.route("/update_details", methods=["GET", "POST"])
@login_required
def update_details():
    if request.method == "POST":
        new_username = request.form["username"]
        new_email = request.form["email"]
        new_details = request.form.get("details", "")
        from flask_login import current_user
        user_id = current_user.id  # Fetch the logged-in user's ID
        cursor = mysql.connection.cursor()
        query = "UPDATE users SET username=%s, email=%s, password=%s WHERE
id=%s"
        values = (new_username, new_email, new_details, user_id)
        print(cursor.mogrify(query, values))  # Prints the exact SQL query
        cursor.execute(query, values)
        flash("Details updated successfully!", "success")
        return redirect(url_for("dashboard"))
    return render_template("update_details.html")

# Reset Password
@app.route("/reset_password", methods=["GET", "POST"])
@login_required
def reset_password():
    if request.method == "POST":
        new_password =
bcrypt.generate_password_hash(request.form["password"]).decode("utf-8")
        cursor = mysql.connection.cursor()
        cursor.execute("UPDATE users SET password=%s WHERE id=%s",
(new_password, current_user.id))
        mysql.connection.commit()
        cursor.close()
        flash("Password reset successful!", "success")
        return redirect(url_for("dashboard"))
    return render_template("reset_password.html")
```

```
# Display Grades (Read-Only)
@app.route("/grades")
@login_required
def grades():
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("SELECT subject, grade FROM grades WHERE user_id=%s",
[current_user.id])
    grades = cursor.fetchall()
    cursor.close()
    return render_template("grades.html", grades=grades)

# Logout
@app.route("/logout")
@login_required
def logout():
    logout_user()
    flash("Logged out successfully.", "info")
    return redirect(url_for("home"))

if __name__ == "__main__":
    app.run(debug=True)
```

HTML CODE

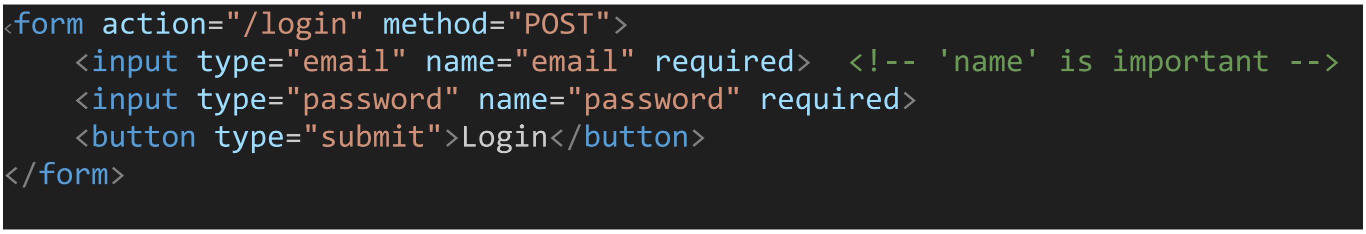
Dashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
</head>
<body>
    <h1>Welcome, {{ username }}</h1>
    <div class="card">
        <div class="card-header">Your Performance</div>
        <div class="card-body">
            <h5>Marks: {{ marks }}</h5>
            <h5>Grade: {{ grade }}</h5>
        </div>
    </div>

    <nav>
        <ul>
            <li><a href="{{ url_for('update_details') }}">Update
Details</a></li>
            <li><a href="{{ url_for('reset_password') }}">Reset
Password</a></li>
```



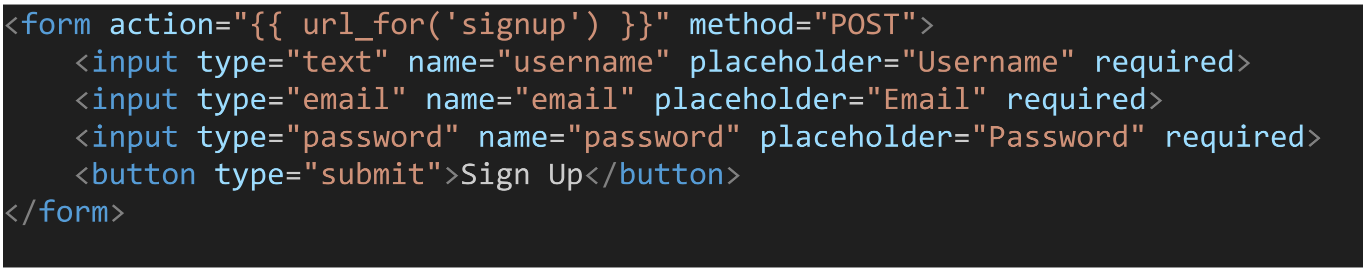
```
            <li><a href="{{ url_for('logout') }}">Logout</a></li>
        </ul>
    </nav>
</body>
</html>
```

Login.html



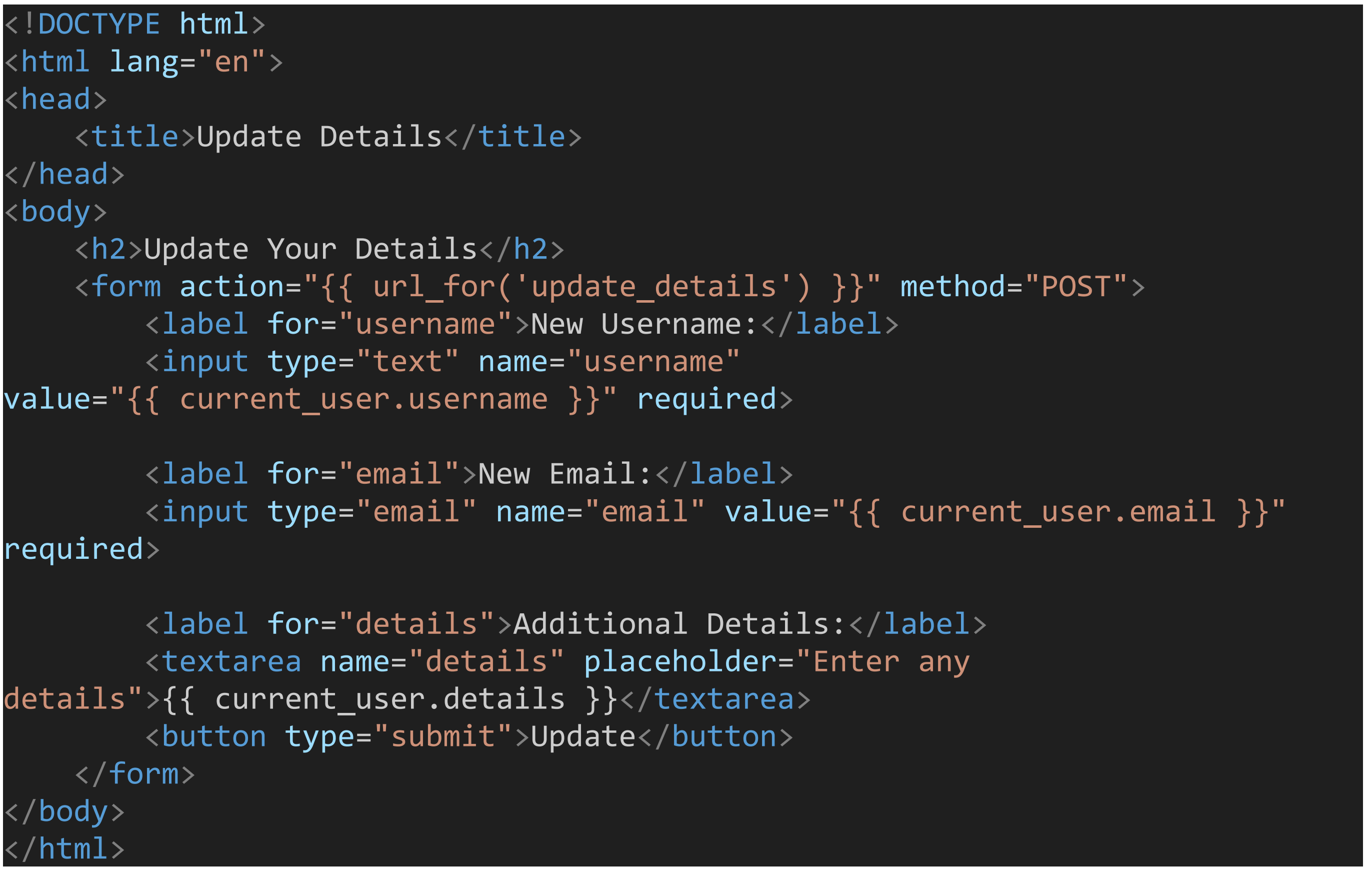
```
<form action="/login" method="POST">
    <input type="email" name="email" required>  <!-- 'name' is important -->
    <input type="password" name="password" required>
    <button type="submit">Login</button>
</form>
```

Signup.html



```
<form action="{{ url_for('signup') }}" 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>
```

Update_details.html



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Update Details</title>
</head>
<body>
    <h2>Update Your Details</h2>
    <form action="{{ url_for('update_details') }}" method="POST">
        <label for="username">New Username:</label>
        <input type="text" name="username"
value="{{ current_user.username }}" required>

        <label for="email">New Email:</label>
        <input type="email" name="email" value="{{ current_user.email }}"
required>

        <label for="details">Additional Details:</label>
        <textarea name="details" placeholder="Enter any
details">{{ current_user.details }}</textarea>
        <button type="submit">Update</button>
    </form>
</body>
</html>
```

Reset_password.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reset Password</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>
<body>
    <nav>
        <a href="{{ url_for('dashboard') }}">Back to Dashboard</a>
        <a href="{{ url_for('logout') }}">Logout</a>
    </nav>
    <div class="container">
        <h2>Reset Password</h2>
        <form method="POST">
            <label for="password">New Password:</label>
            <input type="password" name="password" required>
            <button type="submit">Reset</button>
        </form>
    </div>
</body>
</html>
```

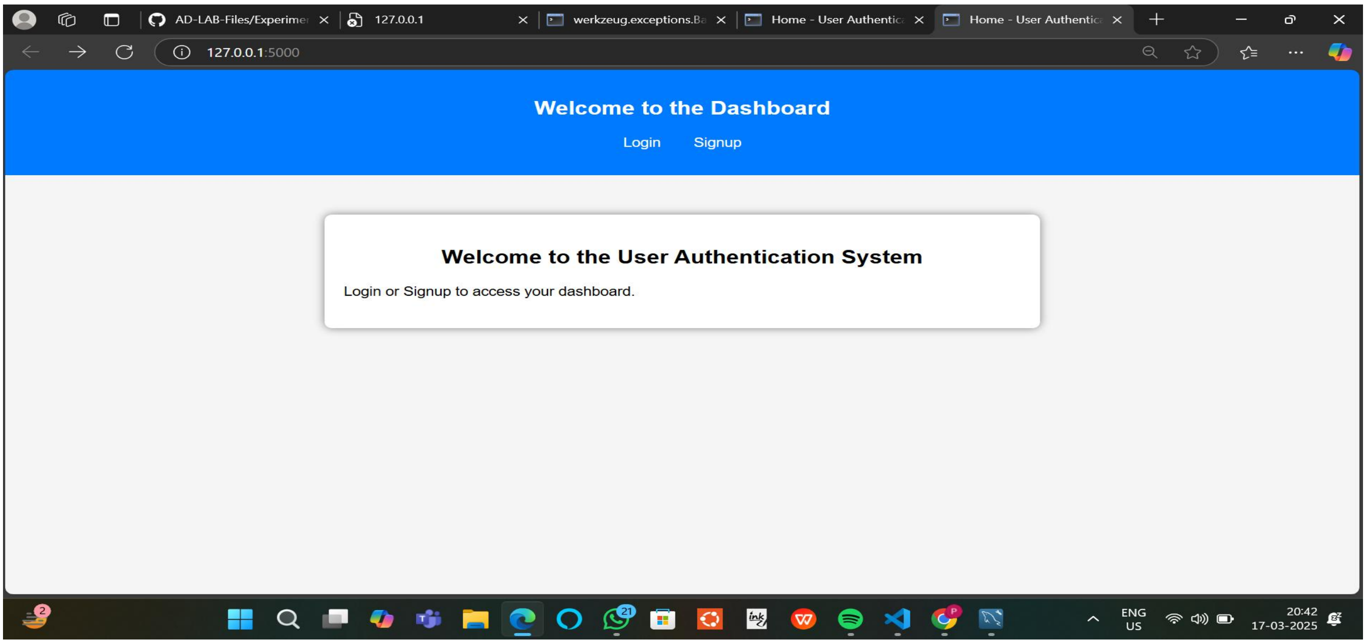
Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - User Authentication</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>
<body>
    <nav>
        <h1>Welcome to the Dashboard</h1>
        <ul>
            {% if session.loggedin %}
                <li><a href="{{ url_for('dashboard') }}">Dashboard</a></li>
                <li><a href="{{ url_for('update_details') }}">Update
Details</a></li>
                <li><a href="{{ url_for('reset_password') }}">Reset
Password</a></li>
                <li><a href="{{ url_for('grades') }}">View Grades</a></li>
                <li><a href="{{ url_for('logout') }}">Logout</a></li>
            {% else %}
                <li><a href="{{ url_for('login') }}">Login</a></li>
                <li><a href="{{ url_for('signup') }}">Signup</a></li>
            {% endif %}
        </ul>
    </nav>
    <div class="container">
        <h2>Welcome to the User Authentication System</h2>
        <p>Login or Signup to access your dashboard.</p>
    </div>
</body>
</html>
```

SQL FILE

```
CREATE DATABASE user_auth;
USE user_auth;

CREATE TABLE users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(100) UNIQUE NOT NULL,
   email VARCHAR(100) UNIQUE NOT NULL,
   password VARCHAR(255) NOT NULL
);

CREATE TABLE grades (
   id INT AUTO_INCREMENT PRIMARY KEY,
   user_id INT NOT NULL,
   subject VARCHAR(100),
   grade VARCHAR(10),
   FOREIGN KEY (user_id) REFERENCES users(id)
);
```

3. Results/Output:- Entire Screen Shot including Date & Time

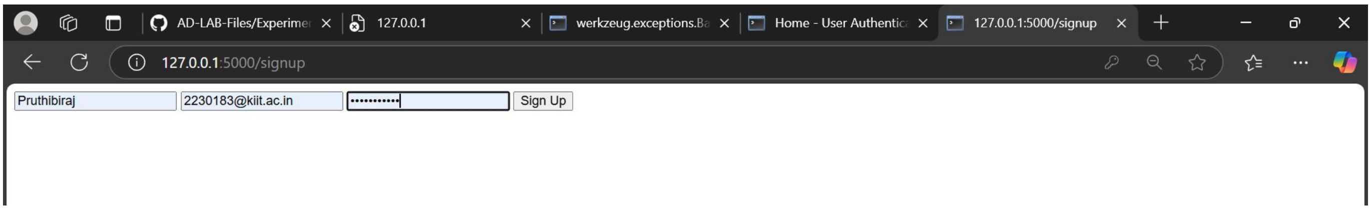


Welcome to the Dashboard

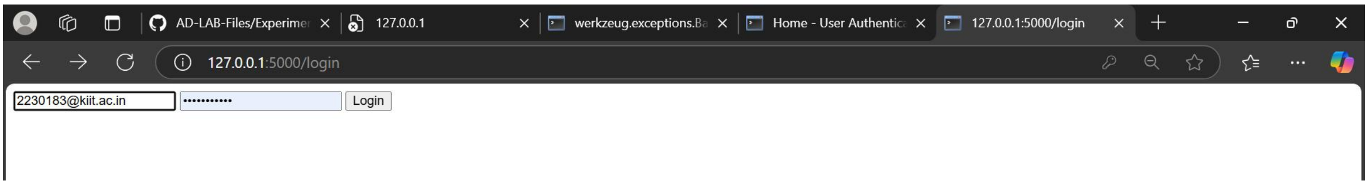
Login Signup

Welcome to the User Authentication System

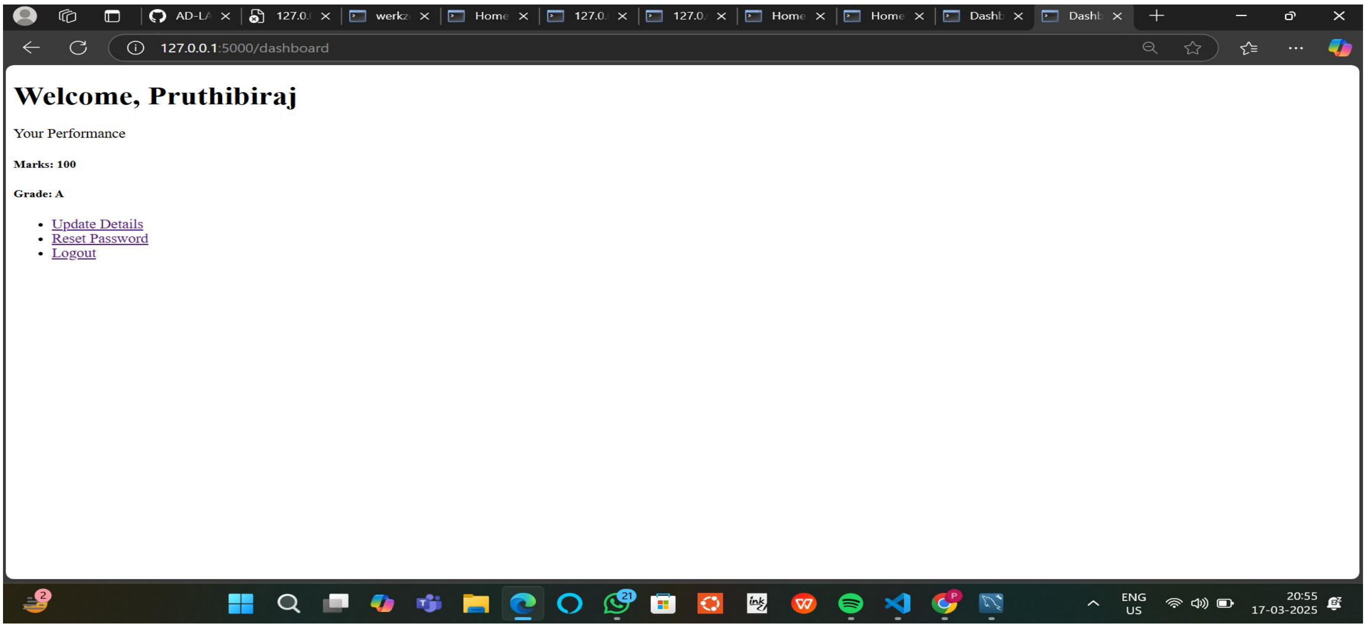
Login or Signup to access your dashboard.



Pruthibiraj | 2230183@kiit.ac.in | •••••••••• | Sign Up



2230183@kiit.ac.in | •••••••••• | Login



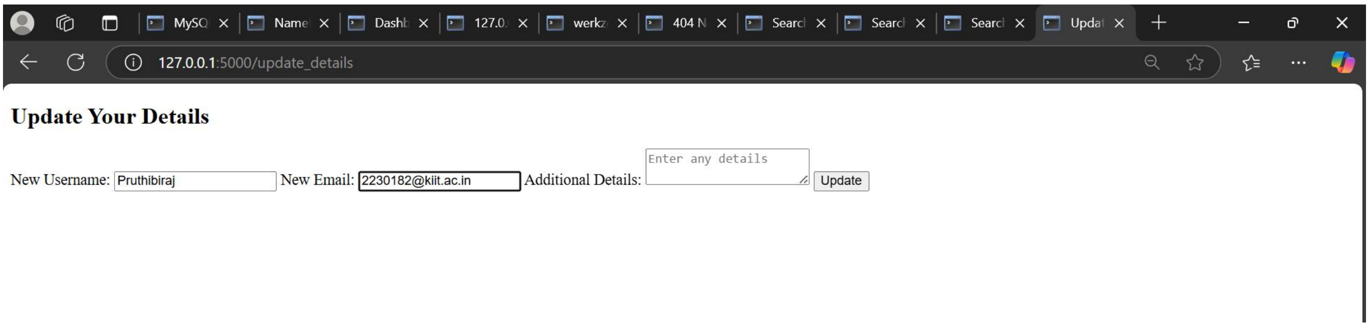
Welcome, Pruthibiraj

Your Performance

Marks: 100

Grade: A

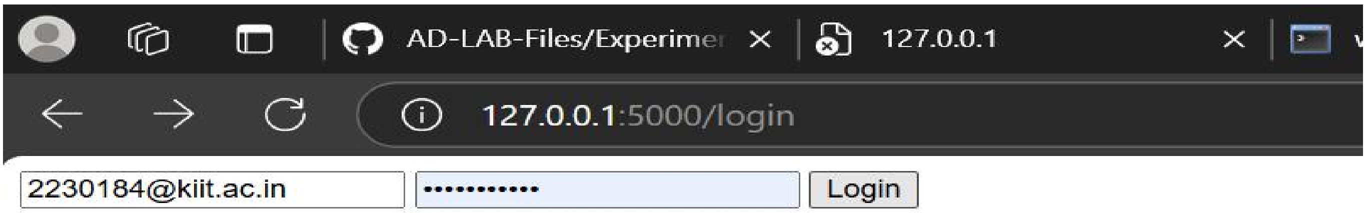
- Update Details
- Reset Password
- Logout



127.0.0.1:5000/update_details

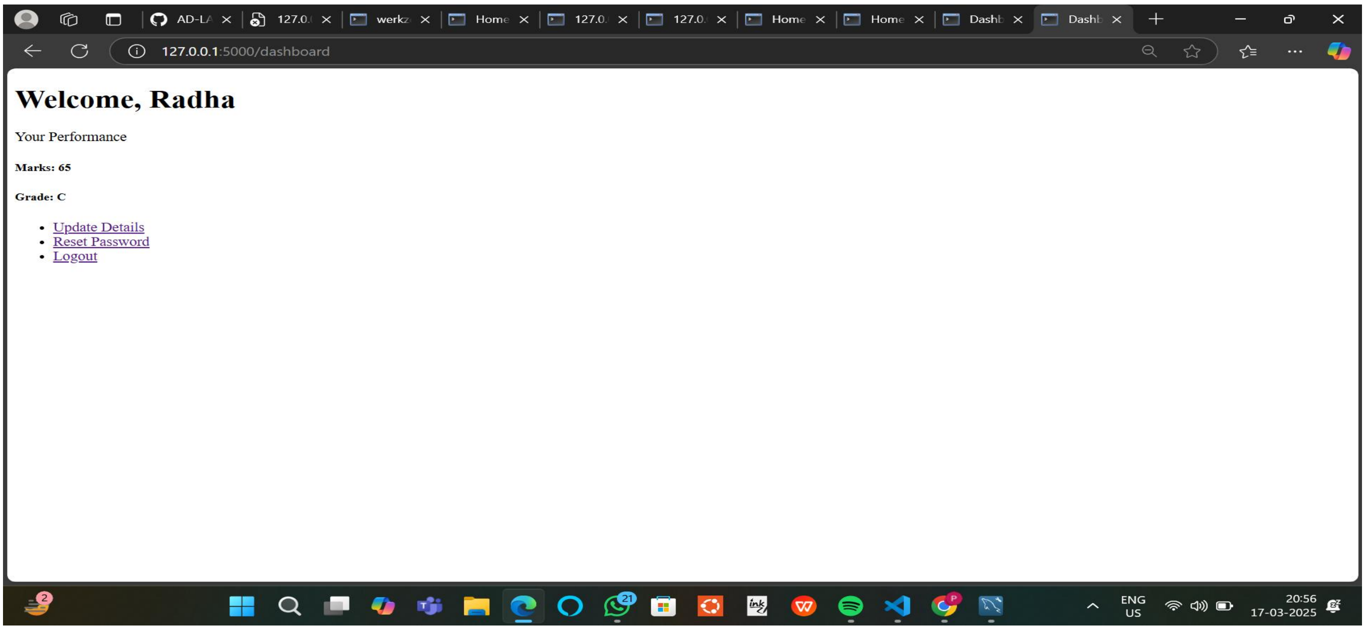
Update Your Details

New Username: Pruthibiraj New Email: 2230182@kiit.ac.in Additional Details: Enter any details Update



127.0.0.1:5000/login

2230184@kiit.ac.in ••••••••••• Login



127.0.0.1:5000/dashboard

Welcome, Radha

Your Performance

Marks: 65

Grade: C

- Update Details
- Reset Password
- Logout

4. Remarks:-

In this experiment, we successfully implemented a user authentication system in Flask with MySQL integration, enabling users to register, log in, and update their details securely. We encountered and resolved key issues, including undefined variables, incorrect table schema, and session handling in Flask-Login. Debugging techniques such as printing SQL queries, verifying session persistence, and manually testing queries in MySQL were crucial in ensuring smooth functionality. This experiment provided hands-on experience in managing user data, handling database transactions, and troubleshooting common operational errors in a web application.

Pruthibiraj Nayak (2230183)

______________________ ____________________________________

(Name of the Student) (Name of the Coordinator)