

LABORATORY REPORT

**Application Development Lab
(CS33002)**

B. Tech Program in ECSc

Submitted By

Name:- Akriti Patro

Roll No: 2230232



**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	Build a Resume using HTML/CSS	07/01/25	13/01/25	
2	Machine Learning for Cat and Dog Classification	14/01/25	20/01/25	
3	Regression Analysis for Stock Prediction	21/01/25	27/01/25	
4	Conversational Chatbot with Any Files	28/01/25	09/02/25	
5	Web Scraper using LLMs	11/02/25	10/03/25	
6	Database Management Using Flask	20/02/25	10/03/25	
7				
8				
9	Open Ended 1			
10	Open Ended 2			

Experiment Number	6
Experiment Title	Database Management Using Flask
Date of Experiment	20/02/25
Date of Submission	10/03/25

1. Objective:- To develop an application for user authentication and document sharing.

2. Procedure:-

1. I installed MySQL Workbench on my system and also installed the flask-mysqldb package to connect my Flask application with MySQL
2. I created a MySQL database where I store user details, including usernames and passwords, to manage authentication securely.
3. I implemented a user authentication and registration system using Flask and MySQL. New users can create an account using the register button, while existing users can log in using their credentials. I also ensured that passwords are securely hashed before storing them in the database.
4. Once logged in, users can update their personal details and reset their passwords through their update profile and reset password settings respectively. This allows them to keep their information up to date.
5. Inside the dashboard, users can view their grades, but they cannot edit them manually.
6. Finally, I built a fully responsive frontend for smooth user interaction. I used HTML, CSS, and Bootstrap to make the interface clean and user-friendly, ensuring it works well on different screen sizes.

3. Code:-

database.py

```

from flask import Flask, render_template, request, redirect, url_for, flash, session, send_file
from flask_bcrypt import Bcrypt
from flask_mysqldb import MySQL
from flask_login import LoginManager, UserMixin, login_user, logout_user, login_required,
current_user
import os
import secrets
print(secrets.token_hex(16))

# Initialize Flask app
app = Flask(__name__)
app.config.from_object('config.Config')

# Initialize extensions
mysql = MySQL(app)
bcrypt = Bcrypt(app)
login_manager = LoginManager(app)
login_manager.login_view = 'login'

# Ensure upload folder exists
if not os.path.exists(app.config['UPLOAD_FOLDER']):
    os.makedirs(app.config['UPLOAD_FOLDER'])

```

```

ALLOWED_EXTENSIONS = {'pdf', 'docx', 'txt'}

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# User class
class User(UserMixin):
    def __init__(self, user_id, username, email):
        self.id = user_id
        self.username = username
        self.email = email

@login_manager.user_loader
def load_user(user_id):
    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM users WHERE user_id = %s", (user_id,))
    user = cur.fetchone()
    cur.close()
    if user:
        return User(user[0], user[1], user[2])
    return None

@app.route('/')
def home():
    return redirect(url_for('login'))

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']

        cur = mysql.connection.cursor()
        cur.execute("SELECT * FROM users WHERE email = %s", (email,))
        existing_user = cur.fetchone()

        if existing_user:
            flash('Email already registered! Please login.', 'danger')
            return redirect(url_for('login'))

        hashed_password = bcrypt.generate_password_hash(password).decode('utf-8')
        cur.execute("INSERT INTO users (username, email, password) VALUES (%s, %s, %s)",
        (username, email, hashed_password))
        mysql.connection.commit()
        cur.close()

        flash('Account created successfully!', 'success')
        return redirect(url_for('login'))

    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 user_id, username, email, password FROM users WHERE email = %s",
(email,))
    user = cur.fetchone()
    cur.close()

    if user and bcrypt.check_password_hash(user[3], password):
        login_user(User(user[0], user[1], user[2]))
        flash("Login successful!", "success")
        return redirect(url_for('dashboard'))
    else:
        flash("Invalid email or password!", "danger")

return render_template('login.html')

@app.route('/dashboard')
@login_required
def dashboard():
    cur = mysql.connection.cursor()
    cur.execute("SELECT subject, grade FROM grades WHERE user_id = %s", (current_user.id,))
    grades = cur.fetchall()
    cur.execute("SELECT id, filename FROM documents WHERE user_id = %s", (current_user.id,))
    documents = cur.fetchall()
    cur.close()
    return render_template('dashboard.html', grades=grades, documents=documents)

@app.route('/upload', methods=['POST'])
@login_required
def upload():
    if 'file' not in request.files:
        flash('No file selected!', 'danger')
        return redirect(url_for('dashboard'))

    file = request.files['file']
    if file.filename == "" or not allowed_file(file.filename):
        flash('Invalid file type!', 'danger')
        return redirect(url_for('dashboard'))

    filepath = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
    file.save(filepath)

    cur = mysql.connection.cursor()
    cur.execute("INSERT INTO documents (user_id, filename, filepath) VALUES (%s, %s, %s)",
(current_user.id, file.filename, filepath))
    mysql.connection.commit()
    cur.close()

    flash('File uploaded successfully!', 'success')
    return redirect(url_for('dashboard'))

```

```

@app.route('/download/<int:doc_id>')
@login_required
def download(doc_id):
    cur = mysql.connection.cursor()
    cur.execute("SELECT filename, filepath FROM documents WHERE id = %s AND user_id
= %s", (doc_id, current_user.id))
    doc = cur.fetchone()
    cur.close()

    if doc:
        return send_file(doc[1], as_attachment=True)
    else:
        flash('File not found!', 'danger')
        return redirect(url_for('dashboard'))

@app.route('/update_profile', methods=['GET', 'POST'])
@login_required
def update_profile():
    if request.method == 'POST':
        new_username = request.form['username']
        new_email = request.form['email']

        cur = mysql.connection.cursor()
        cur.execute("UPDATE users SET username = %s, email = %s WHERE user_id = %s",
(new_username, new_email, current_user.id))
        mysql.connection.commit()
        cur.close()
        flash('Profile updated successfully!', 'success')
        return redirect(url_for('dashboard'))
    return render_template('update_profile.html')

@app.route('/reset_password', methods=['GET', 'POST'])
def reset_password():
    if request.method == 'POST':
        email = request.form['email']
        new_password = request.form['new_password']

        cur = mysql.connection.cursor()
        cur.execute("SELECT * FROM users WHERE email = %s", (email,))
        user = cur.fetchone()

        if user:
            hashed_password = bcrypt.generate_password_hash(new_password).decode('utf-8')
            cur.execute("UPDATE users SET password = %s WHERE email = %s", (hashed_password,
email))
            mysql.connection.commit()
            cur.close()
            flash('Password reset successfully! Please log in with your new password.', 'success')
            return redirect(url_for('login'))
        else:
            flash('Email not found. Please enter a registered email.', 'danger')

    return render_template('reset_password.html')

```

```

@app.route('/logout')
@login_required
def logout():
    logout_user()
    flash("Logged out successfully!", "success")
    return redirect(url_for('login'))

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

```

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>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body>
    <div class="container mt-5">
        <h2 class="mb-4">Welcome, {{ current_user.username }}</h2>

        <!-- Flash Messages -->
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{ category }}" alert-dismissible fade show" role="alert">
                        {{ message }}
                        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}

        <!-- Grades Section -->
        <div class="card mb-4">
            <div class="card-header">
                <h3>Your Grades</h3>
            </div>
            <div class="card-body">
                {% if grades %}
                    <ul class="list-group">
                        {% for subject, grade in grades %}
                            <li class="list-group-item d-flex justify-content-between align-items-center">
                                <strong>{{ subject }}</strong>
                                <span class="badge bg-primary">{{ grade }}</span>
                            </li>
                        {% endfor %}
                    </ul>
                {% else %}
                    <p class="text-muted">No grades available.</p>
                {% endif %}
            </div>
        </div>
    </div>

```

```

        {% endif %}
    </div>
</div>

<!-- File Upload Section -->
<div class="card mb-4">
    <div class="card-header">
        <h3>Upload a Document</h3>
    </div>
    <div class="card-body">
        <form method="POST" action="{{ url_for('upload') }}" enctype="multipart/form-data">
            <div class="mb-3">
                <input type="file" name="file" class="form-control">
            </div>
            <button type="submit" class="btn btn-success">Upload</button>
        </form>
    </div>
</div>

<!-- Uploaded Documents Section -->
<div class="card mb-4">
    <div class="card-header">
        <h3>Your Documents</h3>
    </div>
    <div class="card-body">
        {% if documents %}
            <ul class="list-group">
                {% for doc in documents %}
                    <li class="list-group-item d-flex justify-content-between align-items-center">
                        {{ doc[1] }}
                        <a href="{{ url_for('download', doc_id=doc[0]) }}" class="btn btn-sm btn-
primary">Download</a>
                    </li>
                {% endfor %}
            </ul>
        {% else %}
            <p class="text-muted">No documents uploaded yet.</p>
        {% endif %}
    </div>
</div>

<!-- Profile Management Section -->
<div class="card mb-4">
    <div class="card-header">
        <h3>Profile Management</h3>
    </div>
    <div class="card-body">
        <div class="d-flex gap-2">
            <a href="{{ url_for('update_profile') }}" class="btn btn-primary flex-fill">Update
Profile</a>
            <a href="{{ url_for('reset_password') }}" class="btn btn-warning flex-fill">Reset
Password</a>
        </div>
    </div>
</div>

```



```

</div>

<a href="{{ url_for('logout') }}" class="btn btn-danger">Logout</a>
</div>

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

```

login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body>
  <div class="container mt-5">
    <h2 class="mb-4">Login</h2>

    <!-- Flash Messages -->
    {% with messages = get_flashed_messages(with_categories=true) %}
      {% if messages %}
        {% for category, message in messages %}
          <div class="alert alert-{{ category }} alert-dismissible fade show" role="alert">
            {{ message }}
            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
          </div>
        {% endfor %}
      {% endif %}
    {% endwith %}

    <form method="POST" class="border p-4 rounded shadow">
      <div class="mb-3">
        <label class="form-label">Email</label>
        <input type="email" name="email" class="form-control" placeholder="Enter email" required>
      </div>
      <div class="mb-3">
        <label class="form-label">Password</label>
        <input type="password" name="password" class="form-control" placeholder="Enter password"
required>
      </div>
      <button type="submit" class="btn btn-primary">Login</button>
    </form>

    <p class="mt-3">
      Don't have an account? <a href="{{ url_for('register') }}">Register</a>
    </p>
  </div>

```

```

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

```

register.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body>
    <div class="container mt-5">
        <h2 class="mb-4">Register</h2>

        <!-- Flash Messages -->
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{ category }} alert-dismissible fade show" role="alert">
                        {{ message }}
                        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}

        <form method="POST" class="border p-4 rounded shadow">
            <div class="mb-3">
                <label class="form-label">Username</label>
                <input type="text" name="username" class="form-control" placeholder="Enter username"
required>
            </div>
            <div class="mb-3">
                <label class="form-label">Email</label>
                <input type="email" name="email" class="form-control" placeholder="Enter email" required>
            </div>
            <div class="mb-3">
                <label class="form-label">Password</label>
                <input type="password" name="password" class="form-control" placeholder="Enter password"
required>
            </div>
            <button type="submit" class="btn btn-primary">Register</button>
        </form>

        <p class="mt-3">
            Already have an account? <a href="{{ url_for('login') }}">Login</a>
        </p>
    </div>

```

```

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</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="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="bg-light">
    <div class="container mt-5">
        <div class="card shadow-lg mx-auto" style="max-width: 500px;">
            <div class="card-header bg-warning text-dark text-center">
                <h3>Reset Password</h3>
            </div>
            <div class="card-body">
                <!-- Flash Messages -->
                {% with messages = get_flashed_messages(with_categories=true) %}
                {% if messages %}
                    {% for category, message in messages %}
                        <div class="alert alert-{{ category }} alert-dismissible fade show" role="alert">
                            {{ message }}
                            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
                        </div>
                    {% endfor %}
                {% endif %}
                {% endwith %}

                <form method="POST">
                    <div class="mb-3">
                        <label for="email" class="form-label">Your Email</label>
                        <input type="email" id="email" name="email" class="form-control"
value="{{ current_user.email }}" readonly>
                    </div>

                    <div class="mb-3">
                        <label for="new_password" class="form-label">New Password</label>
                        <input type="password" id="new_password" name="new_password" class="form-control"
placeholder="Enter new password" required>
                    </div>

                    <div class="d-grid gap-2">
                        <button type="submit" class="btn btn-success">Reset Password</button>
                        <a href="{{ url_for('dashboard') }}" class="btn btn-secondary">Back to Dashboard</a>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

```

</div>

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

```

update_profile.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Update Profile</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="bg-light">
  <div class="container mt-5">
    <div class="card shadow-lg mx-auto" style="max-width: 500px;">
      <div class="card-header bg-primary text-white text-center">
        <h3>Update Profile</h3>
      </div>
      <div class="card-body">
        <form method="POST">
          <div class="mb-3">
            <label for="username" class="form-label">New Username</label>
            <input type="text" id="username" name="username" class="form-control"
value="{{ current_user.username }}" placeholder="Enter new username">
          </div>

          <div class="mb-3">
            <label for="email" class="form-label">New Email</label>
            <input type="email" id="email" name="email" class="form-control"
value="{{ current_user.email }}" placeholder="Enter new email">
          </div>

          <div class="d-grid gap-2">
            <button type="submit" class="btn btn-success">Update Profile</button>
            <a href="{{ url_for('dashboard') }}" class="btn btn-secondary">Back to Dashboard</a>
          </div>
        </form>
      </div>
    </div>
  </div>

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

```

Database.sql

```

-- Create Database
CREATE DATABASE IF NOT EXISTS user_management;
USE user_management;

```

```

-- Users Table
CREATE TABLE IF NOT EXISTS users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL
);

-- Check All Users
SELECT * FROM users;

-- Documents Table (Linked to Users)
CREATE TABLE IF NOT EXISTS documents (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    filename VARCHAR(255) NOT NULL,
    filepath VARCHAR(255) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);

-- Check All Documents
SELECT * FROM documents;

-- Grades Table
DROP TABLE grades;
CREATE TABLE IF NOT EXISTS grades (
    grade_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    subject VARCHAR(100) NOT NULL,
    grade VARCHAR(5) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id) ON DELETE CASCADE
);

INSERT INTO grades (user_id, subject, grade) VALUES
(1, 'Mathematics', 'A'),
(1, 'Physics', 'B'),
(1, 'Computer Science', 'A'),
(2, 'Mathematics', 'B'),
(2, 'Physics', 'A'),
(2, 'Computer Science', 'B');

-- Check All Grades
SELECT * FROM grades;

-- Check All Tables
SHOW TABLES;

```

4. Results/Output:-



Register

Username

Enter username

Email

Enter email

Password

Enter password

Register

Already have an account? [Login](#)



Login

Email

akritipatro@gmail.com

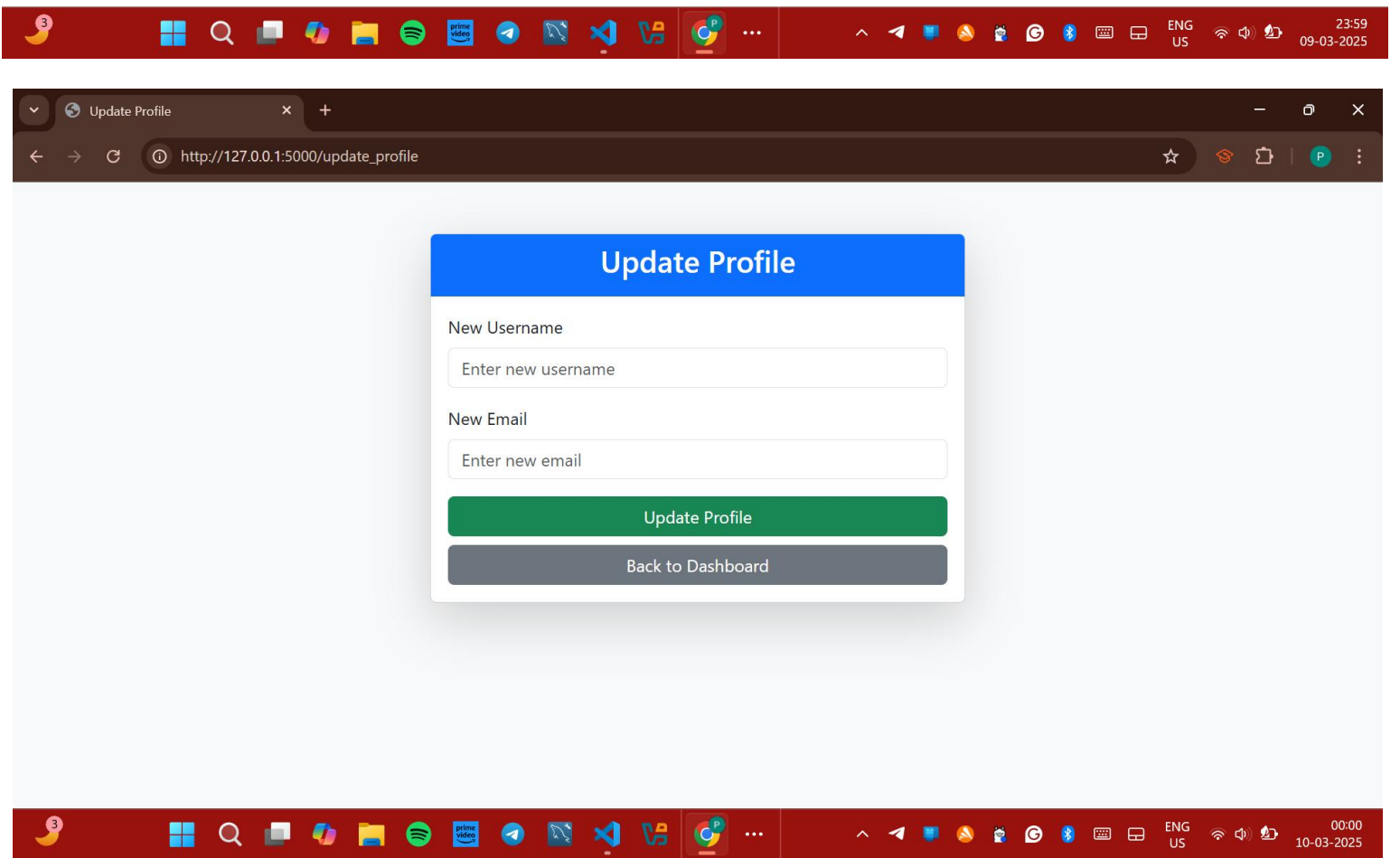
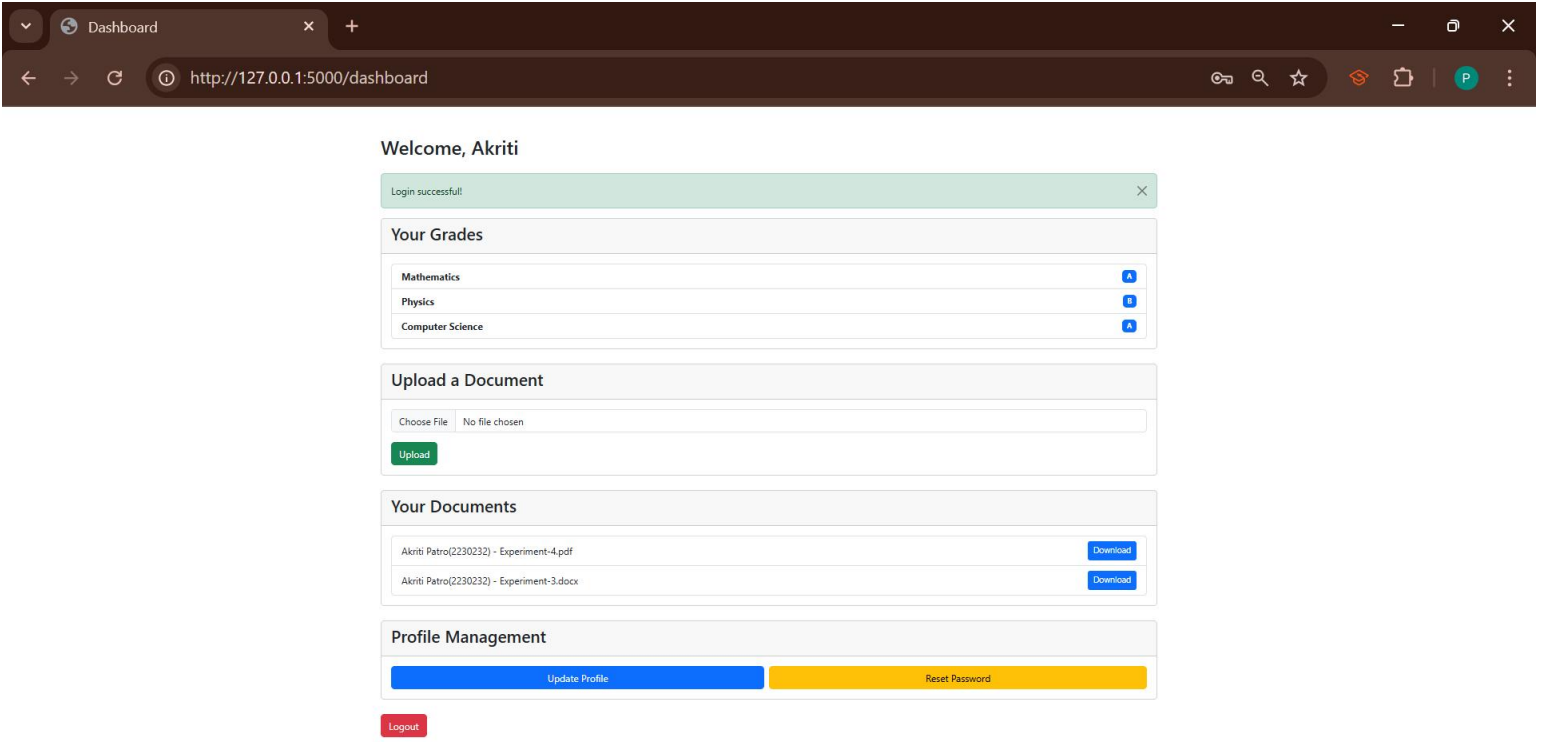
Password

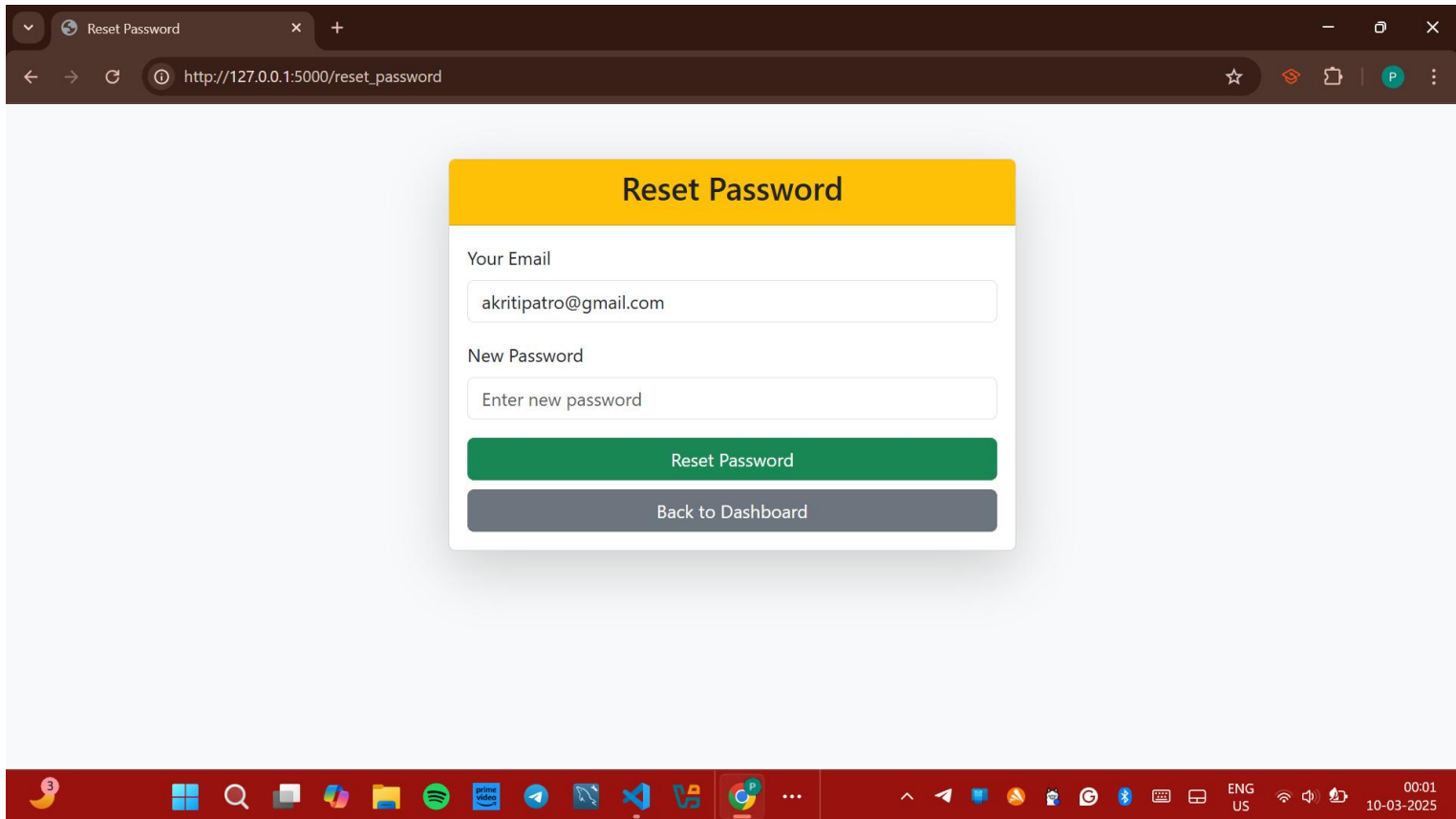
...

Login

Don't have an account? [Register](#)







5. Remarks:-

In this experiment, I developed a database management system using Flask to handle user authentication, profile management, and document storage. I used Flask-MySQLdb to interact with MySQL for storing user credentials, grades, and uploaded documents securely. The Flask backend managed user authentication, allowing new users to register and existing users to log in securely. Users could also update their personal details, reset passwords, and view their grades, which were read-only to maintain data integrity. Additionally, I implemented a file upload system, enabling users to upload and download documents. On the frontend, I built a responsive user interface using HTML, CSS, and Bootstrap. The dashboard displayed user details, grades, and uploaded files while allowing easy navigation between different profile management options.

Signature of the Student

Akriti Patro

Signature of the Lab Coordinator

Prof. Bhargav Appasani