

LABORATORY REPORT
Application Development Lab
(CS33002)

B.Tech Program in ECSc

Submitted By

Name: Shruti Pathak

Roll No: 2230121



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.	To design and develop a professional resume using HTML and CSS.	07/01/2025	13/01/2025	
2.	To design and develop Machine Learning model for Cat and Dog Classification	14/01/2025	20/01/2025	
3.	To perform stock price prediction using Linear Regression and LSTM model	21/01/2025	28/01/2025	
4.	To build a chat-bot capable of answering queries from an uploaded PDF/Word/Excel document.	28/01/25	09/02/25	
5.	Web Scraper using LLMs	19/02/25	17/03/25	
6.	Database Management Using Flask	12/03/25	17/03/25	
7.				
8.				
9.	Open Ended 1			
10.	Open Ended 2			

Experiment Number	6
Experiment Title	Database Management Using Flask
Date of Experiment	12/03/2025
Date of Submission	17/03/2025

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

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.

Code:-

● **app.py:**

```

from flask import Flask, render_template, request, redirect, session,
flash, url_for
from flask_mysqldb import MySQL
from flask_bcrypt import Bcrypt
import MySQLdb.cursors

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


mysql = MySQL(app)
bcrypt = Bcrypt(app)

```

```

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

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

        #  Hash password correctly
        password_hash =
bcrypt.generate_password_hash(password).decode('utf-8')

        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO users (username, email,
password_hash) VALUES (%s, %s, %s)", (username, email,
password_hash))
        mysql.connection.commit()
        cur.close()


        return redirect('/login')

    return render_template('register.html')

# Login Route
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']

        cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cur.execute("SELECT * FROM users WHERE email=%s",
[email])
        user = cur.fetchone()
        cur.close()

```

```
        if user and bcrypt.check_password_hash(user['password_hash'],
password): #  Fixed KeyError
            session['user_id'] = user['id']
            return redirect('/dashboard')
```

```
        flash("Invalid email or password", "danger")
```

```
    return render_template('login.html')
```

```
# Dashboard Route
```

```
@app.route('/dashboard')
```

```
def dashboard():
```

```
    if 'user_id' not in session:
```

```
        return redirect('/login')
```

```
    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
```

```
    # Fetch user details
```

```
        cur.execute("SELECT username, email FROM users WHERE
id=%s", (session['user_id'],))
```

```
        user_data = cur.fetchone()
```

```
    # Fetch documents
```

```
        cur.execute("SELECT filename, file_path FROM documents
WHERE user_id=%s", (session['user_id'],))
```

```
        documents = cur.fetchall()
```

```
    # Fetch grades
```

```
        cur.execute("SELECT id, subject, marks FROM grades WHERE
user_id=%s", (session['user_id'],))
```

```
        grades = cur.fetchall()
```

```
    cur.close()
```

```
    return render_template('dashboard.html', user=user_data,
documents=documents, grades=grades)
```

```
# Update Profile Route
```

```
@app.route('/update_profile', methods=['GET', 'POST'])
```

```
def update_profile():
```

```
    if 'user_id' in session:
```

```
        if request.method == 'POST':
```

```
            name = request.form['name']
```

```

        email = request.form['email']
        user_id = session['user_id']

        cur = mysql.connection.cursor()
        cur.execute("UPDATE users SET username=%s, email=%s
WHERE id=%s", (name, email, user_id))
        mysql.connection.commit()
        cur.close()


        flash("Profile updated successfully!", "success")
        return redirect(url_for('dashboard'))


    cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cur.execute("SELECT * FROM users WHERE id = %s",
[session['user_id']])
    user = cur.fetchone()
    cur.close()

    return render_template("update_profile.html", user=user)

return redirect(url_for('login'))

# Reset Password Route
@app.route('/reset_password', methods=['GET', 'POST'])
def reset_password():
    if 'user_id' in session:
        if request.method == 'POST':
            current_password = request.form['current_password']
            new_password = request.form['new_password']
            confirm_password = request.form['confirm_password']
            user_id = session['user_id']

            cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
            cur.execute("SELECT password_hash FROM users WHERE
id = %s", [user_id]) #  Fixed KeyError
            user = cur.fetchone()
            cur.close()

            if user and
bcrypt.check_password_hash(user['password_hash'], current_password):
#  Fixed password check
                if new_password == confirm_password:

```

```

hashed_password =
bcrypt.generate_password_hash(new_password).decode('utf-8')

        cur = mysql.connection.cursor()
        cur.execute("UPDATE users SET password_hash=%s
WHERE id=%s", (hashed_password, user_id))
        mysql.connection.commit()
        cur.close()

        flash("Password updated successfully!", "success")
        return redirect(url_for('dashboard'))
    else:
        flash("Passwords do not match!", "danger")
    else:
        flash("Current password is incorrect!", "danger")


    return render_template("reset_password.html")

return redirect(url_for('login'))

# View Grades Route
@app.route('/grades')
def view_grades():
    if 'user_id' not in session:
        return redirect('/login')

    cur = mysql.connection.cursor()
    cur.execute("SELECT subject, marks FROM grades WHERE
user_id = %s", (session['user_id'],))
    grades = cur.fetchall()
    cur.close()

    return render_template('grades.html', grades=grades)

# Update Grades Route (Admin Only)
@app.route('/update_grades', methods=['POST'])
def update_grades():
    if 'is_admin' not in session or not session['is_admin']: # 
Admin check fixed
        flash("You do not have permission to edit grades!", "danger")
        return redirect(url_for('view_grades'))

    subject = request.form['subject']

```

```

        new_marks = request.form['marks']
        user_id = request.form['user_id'] # Admin selects which user to
update

        cur = mysql.connection.cursor()
        cur.execute("UPDATE grades SET marks=%s WHERE
user_id=%s AND subject=%s", (new_marks, user_id, subject))
        mysql.connection.commit()
        cur.close()

        flash("Grades updated successfully!", "success")
        return redirect(url_for('view_grades'))

# Run the app
if __name__ == '__main__':
    app.secret_key = 'your_secret_key'
    app.run(debug=True)

```

Reset_password.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Reset Password</title>
</head>
<body>
    <h2>Reset Password</h2>
    <form method="POST">
        <label>Current Password:</label>
        <input type="password" name="current_password" required><br>
        <label>New Password:</label>
        <input type="password" name="new_password" required><br>
        <label>Confirm New Password:</label>
        <input type="password" name="confirm_password" required><br>
        <button type="submit">Reset Password</button>
    </form>
    <a href="{{ url_for('dashboard') }}">Back to Dashboard</a>
</body>
</html>

```

Update_profile.html

```

{% extends "base.html" %}

{% block content %}

```



```
<style>
.update-container {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background-color: #f4f4f4;
}
.update-card {
  background: #fff;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);
  text-align: center;
  width: 40%;
}
.update-form label {
  display: block;
  font-weight: bold;
  margin: 10px 0 5px;
}
.update-form input {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ddd;
  border-radius: 5px;
}
.update-btn {
  background-color: #007bff;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  font-size: 16px;
  margin-top: 10px;
}
.update-btn:hover {
  background-color: #0056b3;
}
.back-link {
  display: block;
```

```

        margin-top: 15px;
        text-decoration: none;
        font-weight: bold;
        color: #007bff;
    }
    .back-link:hover {
        text-decoration: underline;
    }
</style>

<div class="update-container">
    <div class="update-card">
        <h2 class="text-primary">Update Profile</h2>
        <form method="POST" class="update-form">
            <label>Name:</label>
            <input type="text" name="name" value="{{ user.name }}"
required>
            <label>Email:</label>
            <input type="email" name="email" value="{{ user.email }}"
required>
            <button type="submit" class="update-btn">Update
Profile</button>
        </form>
        <a href="{{ url_for('dashboard') }}" class="back-link">Back to
Dashboard</a>
    </div>
</div>
{% endblock %}

```

Dashboard.html

```

{% extends "base.html" %}

{% block content %}
<style>
    .dashboard-container {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        background-color: #f4f4f4;
    }
    .dashboard-card {
        background: #fff;

```

```
padding: 30px;
border-radius: 10px;
box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);
text-align: center;
width: 50%;
}
.grades-table {
width: 100%;
margin-top: 20px;
border-collapse: collapse;
}
.grades-table th, .grades-table td {
padding: 10px;
border: 1px solid #ddd;
text-align: center;
}
.grades-table th {
background-color: #007bff;
color: white;
}
.action-buttons {
margin-top: 20px;
}
.action-buttons a {
text-decoration: none;
padding: 10px 20px;
margin: 10px;
border-radius: 5px;
font-weight: bold;
display: inline-block;
}
.update-btn {
background-color: #28a745;
color: white;
}
.reset-btn {
background-color: #dc3545;
color: white;
}
.update-btn:hover {
background-color: #218838;
}
.reset-btn:hover {
```

```

        background-color: #c82333;
    }
</style>

<div class="dashboard-container">
    <div class="dashboard-card">
        <h2 class="text-primary">Welcome to Your Dashboard, {{ user[0]
    }}!</h2>
        <p class="text-muted">Your Email: <strong>{{ user[1]
    }}</strong></p>

        <hr>

        <h3 class="text-success">Your Grades:</h3>
        {% if grades %}
        <table class="grades-table">
            <thead>
                <tr>
                    <th>Subject</th>
                    <th>Marks</th>
                </tr>
            </thead>
            <tbody>
                {% for grade in grades %}
                <tr>
                    <td>{{ grade[1] }}</td>
                    <td>{{ grade[2] }}</td>
                </tr>
                <script>console.log("Subject:", "{{ grade[0] }}", "Marks:", "{{
grade[1] }}");</script>
                {% endfor %}
            </tbody>
        </table>
        {% else %}
        <p class="text-warning">No grades available.</p>
        {% endif %}

        <div class="action-buttons">
            <a href="{{ url_for('update_profile') }}"
class="update-btn">Update Profile</a>
            <a href="{{ url_for('reset_password') }}" class="reset-btn">Reset
Password</a>

```

```

        </div>
    </div>
</div>
{% endblock %}

```

- **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 Management</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 50px;
        }
        h1 {
            color: #333;
        }
        a {
            display: inline-block;
            margin: 10px;
            padding: 10px 20px;
            text-decoration: none;
            background-color: #007BFF;
            color: white;
            border-radius: 5px;
        }
        a:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <h1>Welcome to User Management System</h1>
    <p>Please choose an option:</p>
    <a href="{ { url_for('register') } }">Register</a>

```

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

Login and register.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Login & Register</title>
    <style>
        /* Basic Styling */
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
            margin: 0;
        }

        .container {
            background: #ffffff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            width: 300px;
            text-align: center;
        }

        h2 {
            margin-bottom: 15px;
            color: #333;
        }

        /* Input Fields */
        input {
            width: 90%;
            padding: 10px;
            margin: 8px 0;
            border: 1px solid #ccc;
```

```

        border-radius: 5px;
        font-size: 14px;
    }

    /* Button Styling */
    button {
        width: 100%;
        padding: 10px;
        border: none;
        border-radius: 5px;
        background: #007BFF;
        color: white;
        font-size: 16px;
        cursor: pointer;
        margin-top: 10px;
    }

    button:hover {
        background: #0056b3;
    }

    .divider {
        margin: 15px 0;
        color: #777;
    }
</style>
</head>
<body>

    <div class="container">
        <h2>Login</h2>
        <form method="POST">
            <input type="email" name="email" placeholder="Email"
required>
                <input type="password" name="password"
placeholder="Password" required>
                <button type="submit">Login</button>
        </form>
    <h2>Register</h2>
    <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">Register</button>
    </form>

</div>

</body>
</html>
```

Results/Output:

Welcome to User Management System

Please choose an option:

Register

Login

Login

sp.18pathak@gmail.com

.....

Login

Welcome to Your Dashboard, Shruti Pathak!

Your Email: sp.18pathak@gmail.com

Your Grades:

Subject	Marks
Math	85
Science	90
Math	85
Science	90

Update Profile

Reset Password

Update Profile

Name:

Chirkut

Email:

sp.18pathak@gmail.com

Update Profile

[Back to Dashboard](#)

Welcome to Your Dashboard, Chirkut!

Your Email: sp.18pathak@gmail.com

Your Grades:

Subject	Marks
Math	85
Science	90
Math	85
Science	90

[Update Profile](#)[Reset Password](#)

Remarks:- This experiment focuses on developing a user authentication and document-sharing application using Flask and MySQL. It involves setting up MySQL Workbench and the Flask-MySQLdb package for database integration. The system allows new users to register via a signup form, while existing users can log in with their credentials. Users can update personal details, reset passwords, and view their grades, though editing grades is restricted. A responsive frontend is designed for seamless user interactions, ensuring a smooth and secure experience for managing authentication and shared documents.

Signature of the Student

Shruti Pathak

Signature of the Lab Coordinator
