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	11/03/25	
6	Database Management Using Flask	20/02/25	17/03/25	
7				
8				
9	Open Ended 1			
10	Open Ended 2			

Experiment Number	
	Database Management Using Flask
Date of Experiment	20/02/25
Date of Submission	17/03/25

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 berypt import Berypt
      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 berypt.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 %}
                class="list-group-item d-flex justify-content-between align-items-center">
                  <strong>{{ subject }}</strong>
                  <span class="badge bg-primary">{{ grade }}</span>
                </1i>
              {% endfor %}
           {% else %}
           No grades available.
```

```
{% 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 %}
           {% for doc in documents %}
                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>
               {% endfor %}
           {% else %}
           No documents uploaded yet.
         {% endif %}
      </div>
    </div>
    <!-- Profile Management Section -->
    <div class="card mb-4">
      <div class="card-header">
         <h3>Profile Management</h3>
      <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>
    <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>
  k 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">
              <button type="button" class="btn-close" data-bs-dismiss="alert" aria-</pre>
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>
       <button type="submit" class="btn btn-primary">Login</button>
    </form>
    Don't have an account? <a href="{{ url_for('register') }}">Register</a>
```

</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>
  k 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">
              <button type="button" class="btn-close" data-bs-dismiss="alert" aria-</pre>
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"</p>
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>
    Already have an account? <a href="\{\{ url_for('login') \}\}">Login</a>
    </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>
  k 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-</pre>
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"</pre>
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"</pre>
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>
<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"</pre>
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"</pre>
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

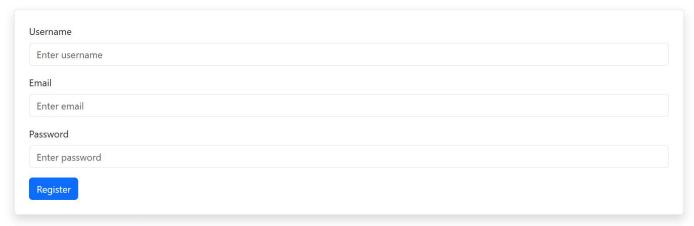
```
-- Create Database
CREATE DATABASE IF NOT EXISTS user_management;
USE user_management;
```

```
-- 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



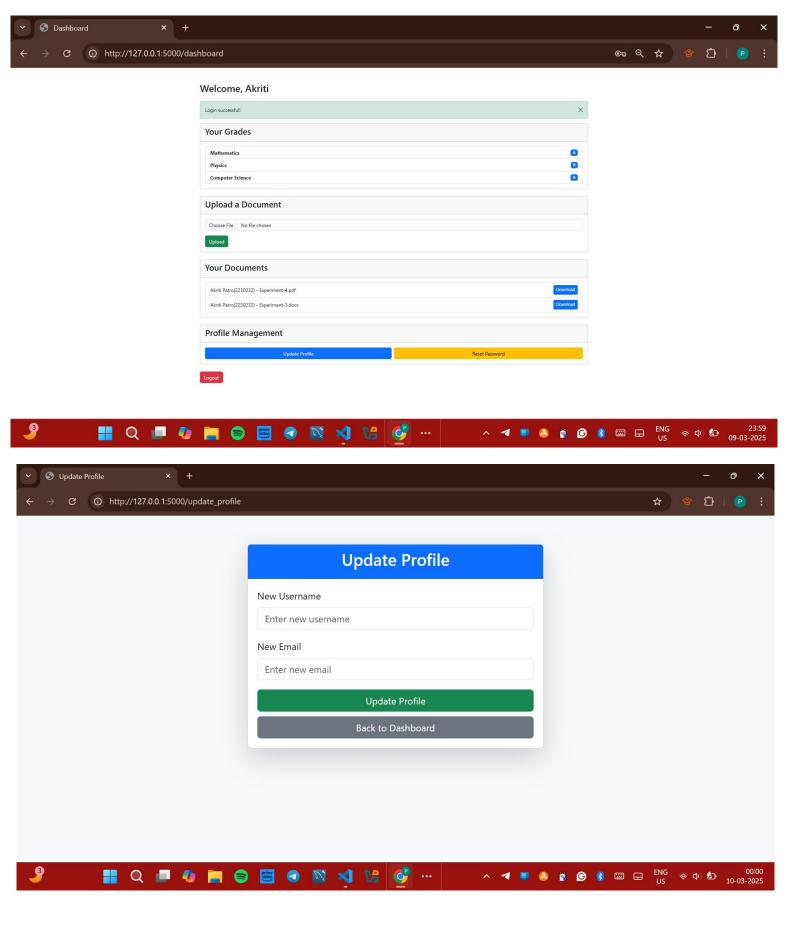
Already have an account? Login

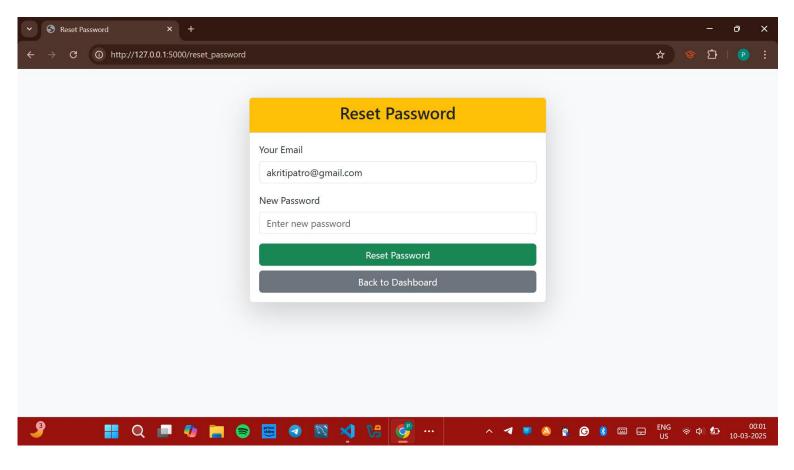




Don't have an account? Register







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