MINI PROJECT

ONLINE TEST PORTAL

DEPARTMENT : INFORMATION TECHNOLOGY

COURSE NAME : DATABASE SYSTEMS LABORATORY

COURSE CODE : 22IPC408

PREPARED BY,

SWARNAMBIKA K [2303717720522053]

NAVYAA SHARMA [2303717720522059]

RASEETHA FATHIMA K [2303717720522304]

SRISATHYA M [2303717720522306]

AIM :

The aim of this project is to develop a database-driven Online Test Portal that enables students to take tests online and allows administrators to manage tests and evaluate results efficiently.This system aims to simplify the test process, ensure automatic result generation, and maintain secure and structured storage of data using a relational database.It also seeks to demonstrate the practical application of database design, ER modeling, SQL queries, and data flow management within a real-world educational context.

OBJECTIVES :

- To design and implement a secure Online Test Portal
- To create a relational database to manage tests, users, and results
- To enable automated question evaluation for MCQs
- To allow admin to create, update and delete tests/questions
- To generate performance reports for students

SCOPE :

This project focuses on online test management for multiple users (Admin and Students). It allows test creation, question management, real-time participation, and automatic evaluation.The portal is suitable for educational institutions, training centers, or companies conducting online assessments.Subjective evaluations and external integrations (like emails or SMS alerts) are not included in the current scope.

SYSTEM REQUIREMENTS :

- **HARDWARE REQUIREMENTS :**

Component	**Specification**
Processor	Intel Core i3 or above
RAM	4 GB or higher
Hard Disk	Minimum 500 MB free space

Monitor	Standard HD Display
Input Devices	Keyboard and Mouse
Network	Internet connection (for real-time testing)

- **SOFTWARE REQUIREMENTS :**

Software	**Version / Details**
Operating System	Windows 10 / Linux / macOS
Frontend Technologies	HTML, CSS
Backend Framework	Python Flask
Database	MySQL
Server	Flask development server
IDE / Code Editor	VS Code / PyCharm / Sublime Text
Web Browser	Google Chrome / Mozilla Firefox
Optional Tools	DB Browser for SQLite (GUI)

MODULES DESCRIPTION :

1. User Authentication Module

- Handles login and registration for both Admin and Students.
- Verifies user credentials using data stored in the database.
- Provides session-based access to protect pages.

2. Admin Module

- Allows the admin to:
- Create and manage tests
- Add, update, or delete questions
- View test reports and student performance
- Admin has full control over the content and users.

3. Student Module

- Allows students to:
- Login and view available tests

- Take tests within the time limit
- View their test results and scores
- Ensures that only valid users can access tests.

4. Test Management Module
    - Manages the creation and scheduling of tests.
    - Handles setting test duration, number of questions, and topics.
    - Links questions to specific tests and subjects.
5. Evaluation & Result Module
    - Automatically checks answers for MCQ-type questions.
    - Calculates total score and displays it to the student.
    - Stores results in the database for admin review.
6. Database Management Module
    - Manages all data storage using SQLite/MySQL.
    - Maintains tables for users, tests, questions, answers, and results.
    - Ensures data integrity and relational consistency.

PROJECT STRUCTURE :

```
        online_test_portal/
 ├── app.py
 ├── templates/
│   ├── layout.html
│   ├── login.html
│   ├── register.html
│   ├── dashboard.html
│   ├── create_test.html
│   ├── take_test.html
```

```
|   └── result.html
├── static/
|   └── style.css
└── database/
    └── schema.sql
```

DATABASE SCHEMA (MySQL) :

```
-- Users table
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(150) NOT NULL UNIQUE,
    email VARCHAR(150) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    role ENUM('admin', 'student') NOT NULL
);

-- Tests table
CREATE TABLE tests (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    subject VARCHAR(255) NOT NULL,
    duration INT NOT NULL,
    created_by INT,
    FOREIGN KEY (created_by) REFERENCES users(id)
);
```

```sql
-- Questions table
CREATE TABLE questions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    test_id INT,
    question_text TEXT NOT NULL,
    option_a VARCHAR(255),
    option_b VARCHAR(255),
    option_c VARCHAR(255),
    option_d VARCHAR(255),
    correct_option CHAR(1),
    FOREIGN KEY (test_id) REFERENCES tests(id)
);

-- Results table
CREATE TABLE results (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    test_id INT,
    score INT,
    total_questions INT,
    date_taken DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (test_id) REFERENCES tests(id)
);
```

BACKEND: app.py

```
from flask import Flask, render_template, request, redirect, session, url_for

from flask_mysqldb import MySQL

import MySQLdb.cursors

import re

from werkzeug.security import generate_password_hash,
check_password_hash


app = Flask(_name_)

app.secret_key = 'your_secret_key'


# MySQL configurations

app.config['MYSQL_HOST'] = 'localhost'

app.config['MYSQL_USER'] = 'root'

app.config['MYSQL_PASSWORD'] = 'your_mysql_password'

app.config['MYSQL_DB'] = 'online_test_portal'


mysql = MySQL(app)


@app.route('/')

def home():

    return render_template('login.html')


# Add routes for register, login, dashboard, create_test, take_test, and result
here


if _name_ == '_main_':
    app.run(debug=True)
```

FRONTEND TEMPLATES :

templates/layout.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Online Test Portal</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <header>
        <h1>Online Test Portal</h1>
    </header>
    <main>
        {% block content %}{% endblock %}
    </main>
</body>
</html>
```

templates/login.html

```
{% extends "layout.html" %}
{% block content %}
<h2>Login</h2>
<form method="POST" action="{{ url_for('login') }}">
```

```
    <label>Username:</label><input type="text" name="username"
required><br>
    <label>Password:</label><input type="password" name="password"
required><br>
    <button type="submit">Login</button>
</form>
<p>Don't have an account? <a href="{{ url_for('register') }}">Register
here</a></p>
{% endblock %}
```

APPLICATION TESTING

1.Set Up Virtual Environment:

```
python -m venv venv
source venv/bin
```

2.Install Dependencies:

```
pip install flask flask-mysqldb werkzeug
```

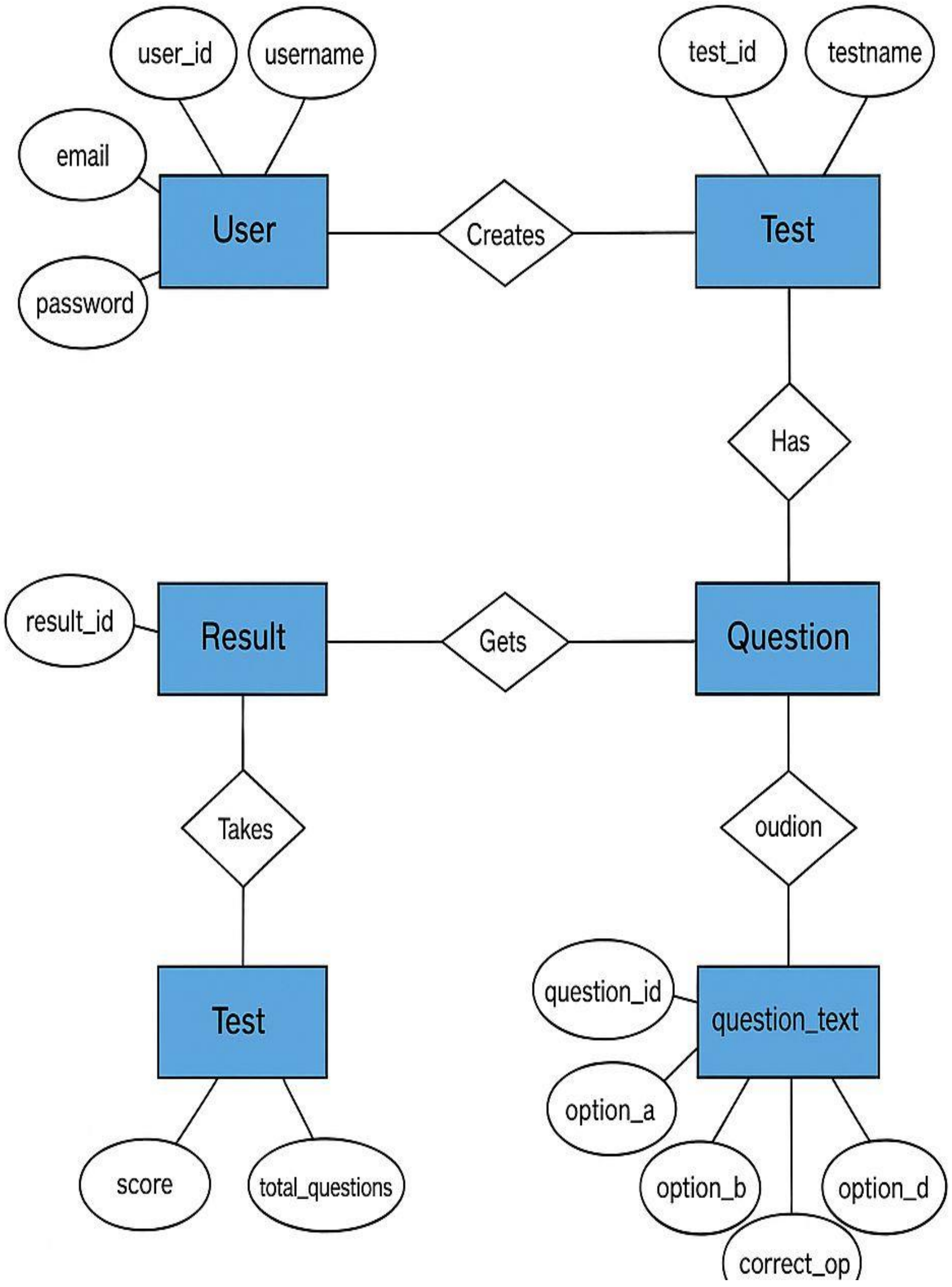
3.Run the Application:

```
python app.py
```

4.Access the Portal:

Open your browser and navigate to http://localhost:5000.

ER DIAGRAM :


user_id
username
email
password
User
Creates
Test
test_id
testname
Has
Result
result_id
Gets
Question
Takes
oudion
Test
score
total_questions
question_id
question_text
option_a
option_b
correct_op
option_d

SYSTEMS ARCHITECTURE EXPLANATION :

1. Presentation Layer (Frontend):

- Built using HTML/CSS
- Responsible for user interaction
- Pages: Login, Register, Dashboard, Test Page, Result Page

2. Application Layer (Backend):

- Built with Python Flask
- Handles:
- Routing (URL to function mapping)
- Form validation
- Test logic and result evaluation

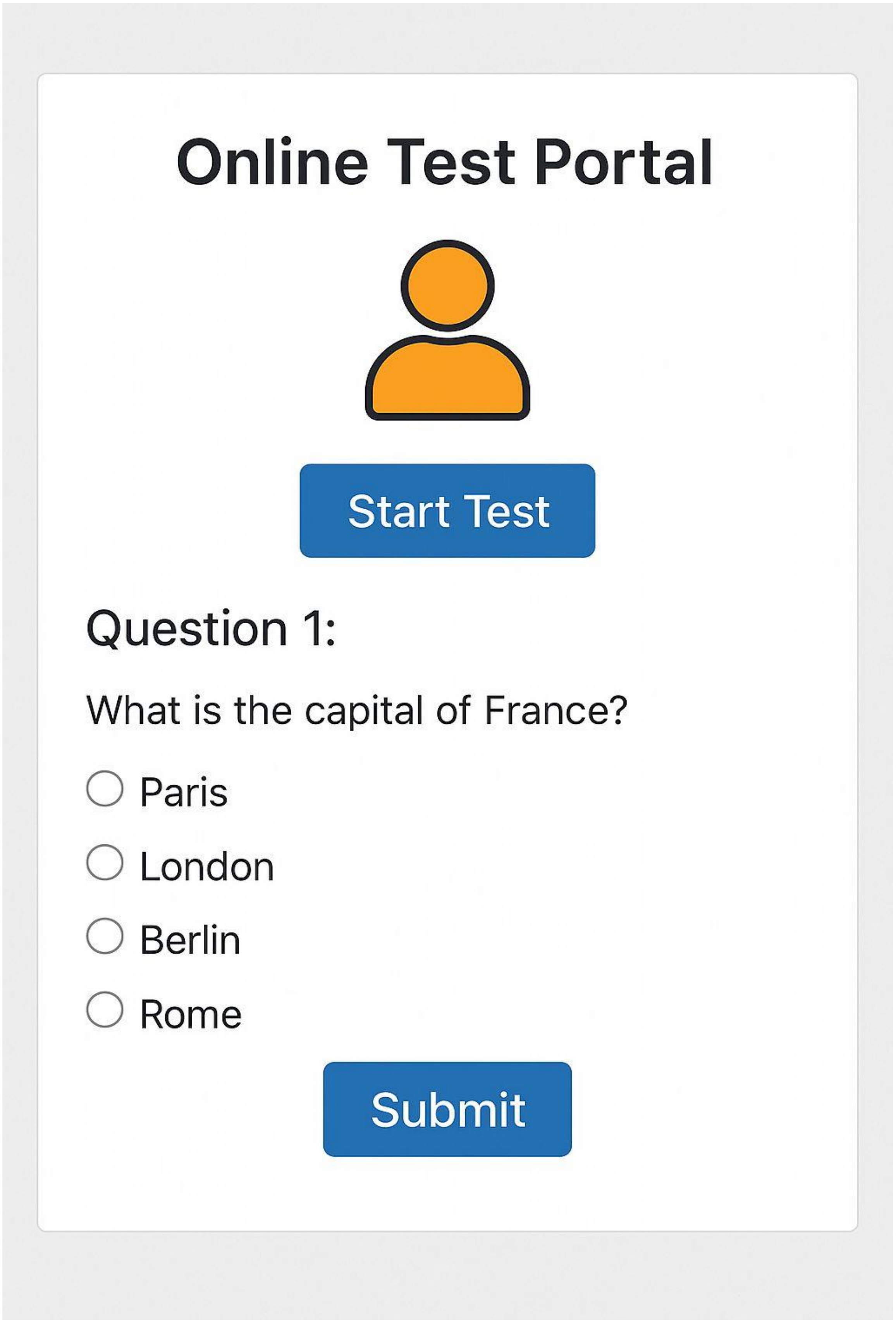
3. Data Layer (Database):

- Managed by MySQL
- Stores:
- User data
- Test metadata
- Questions
- Results

ARCHITECTURE DIAGRAM :

[User Browser] → [HTML/CSS (Frontend)] → [Flask (Python Backend)] → [MySQL Database]

OUTPUT :



Online Test Portal

Start Test

Question 1:

What is the capital of France?

◯ Paris

◯ London

◯ Berlin

◯ Rome

Submit

ADVANTAGES OF THE PROJECT :

- Easy access to tests from anywhere
- Time-efficient and paperless evaluation
- Real-time result generation
- User-friendly interface

LIMITATIONS :

- Limited to objective questions only
- No webcam monitoring or cheating prevention
- Requires internet connection

FUTURE ENHANCEMENTS :

- Add subjective question handling
- Implement timer and auto-submit
- Add admin panel for question management
- Use authentication and session management

CONCLUSION :

The Online Test Portal serves as a simple yet effective platform for conducting quizzes and tests digitally. It demonstrates integration of frontend, backend, and database, fulfilling the project goal of a complete web-based test system.