Prompt:

Online Examination System Problem Statement Traditional examination systems often require significant manual effort for question paper preparation, answer evaluation, and result processing. This leads to inefficiencies, delays, and human errors in grading. The Online Examination System aims to automate these processes, ensuring accuracy, efficiency, and accessibility. By leveraging technology, it provides a centralized platform for managing question banks, conducting exams securely, and instantly grading student responses, thus reducing administrative burdens and enhancing the examination experience for all stakeholders. Introduction The Online Examination System is a database management system designed to facilitate the seamless administration of exams, ensuring efficient question management, student answer submissions, and automated grading. The system assigns a unique Exam ID to each test, allowing students to participate in various exams effortlessly. It reduces manual efforts, minimizes errors in evaluation, and provides a user-friendly interface for students, faculty, and administrators to manage examination-related activities. Project Modules

1. Student Module • Features: Register for exams and receive a unique Student ID. Using this ID, view available exams and participate in them. Submit answers online, check exam results, and review performance reports. Additionally, update personal details, such as contact information, whenever necessary. • Entities: The Student module stores a student’s name, RegNo, StudentID, and contact info. The Exam module records the ExamID, subject, date, time, and Total\_marks. The Answer module stores StudentID, ExamID, and Student\_responses.
2. Faculty Module • Features: Create and manage question banks, schedule exams, assign exams to students, and oversee student participation. The module also facilitates manual review of subjective answers when necessary and allows faculty to generate performance reports. • Entities: The Faculty module stores faculty details (FacultyID, Name, Contact Info), while the QuestionBank module records questions with QuestionID, ExamID, Subject, and Marks.
3. Admin Module • Features: Manage student and faculty accounts, oversee examinations, ensure system security, generate reports, and handle system logs to track activities for auditing purposes. • Entities: The Admin module stores admin details (AdminID, Name, Contact Info), while the SystemLogs module records system activities with log ID, timestamp, and activity details. Additional Entities: The College (CollegeID, Name, Address) module stores college details, the Venue (VenueID, VenueName, Capacity) module stores venue details for offline exams if applicable, and the Schedule (ScheduleID, ExamID, Date, Time) module manages examination schedules. Software Requirements • Frontend: Basic JS, HTML and CSS for simple interface design. • Backend: PHP or Python (Flask) for minimal backend logic. • Database: MySQL (primary focus for database management). • Tools: MySQL (for database setup), Visual Studio Code (for coding).

Give me 100% errorfree codes of the full project end to end that we need and also how to compile and run it.

Reply:

Creating a complete Online Examination System from scratch is a significant task that involves multiple components, including frontend, backend, and database management. Below, I will provide a simplified version of the project, including the necessary code snippets for each module. This will not be a fully-fledged application but will give you a solid foundation to build upon.

**Project Structure**

RunCopy code

1OnlineExaminationSystem/

2│

3├── frontend/

4│ ├── index.html

5│ ├── style.css

6│ └── script.js

7│

8├── backend/

9│ ├── config.php

10│ ├── student.php

11│ ├── faculty.php

12│ ├── admin.php

13│ └── db.php

14│

15└── database/

16 └── schema.sql

**1. Database Setup**

Create a file named **schema.sql** in the **database** folder to set up the database schema.

sql

RunCopy code

1CREATE DATABASE OnlineExaminationSystem;

2

3USE OnlineExaminationSystem;

4

5CREATE TABLE Students (

6 StudentID INT AUTO\_INCREMENT PRIMARY KEY,

7 Name VARCHAR(100),

8 RegNo VARCHAR(50),

9 ContactInfo VARCHAR(100)

10);

11

12CREATE TABLE Faculty (

13 FacultyID INT AUTO\_INCREMENT PRIMARY KEY,

14 Name VARCHAR(100),

15 ContactInfo VARCHAR(100)

16);

17

18CREATE TABLE Admin (

19 AdminID INT AUTO\_INCREMENT PRIMARY KEY,

20 Name VARCHAR(100),

21 ContactInfo VARCHAR(100)

22);

23

24CREATE TABLE Exams (

25 ExamID INT AUTO\_INCREMENT PRIMARY KEY,

26 Subject VARCHAR(100),

27 Date DATE,

28 Time TIME,

29 TotalMarks INT

30);

31

32CREATE TABLE Answers (

33 AnswerID INT AUTO\_INCREMENT PRIMARY KEY,

34 StudentID INT,

35 ExamID INT,

36 StudentResponses TEXT,

37 FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

38 FOREIGN KEY (ExamID) REFERENCES Exams(ExamID)

39);

40

41CREATE TABLE QuestionBank (

42 QuestionID INT AUTO\_INCREMENT PRIMARY KEY,

43 ExamID INT,

44 Subject VARCHAR(100),

45 Marks INT,

46 Question TEXT,

47 FOREIGN KEY (ExamID) REFERENCES Exams(ExamID)

48);

49

50CREATE TABLE SystemLogs (

51 LogID INT AUTO\_INCREMENT PRIMARY KEY,

52 Timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

53 ActivityDetails TEXT

54);

**2. Backend Code**

**config.php**

php

RunCopy code

1<?php

2$host = 'localhost';

3$db = 'OnlineExaminationSystem';

4$user = 'root'; // Change this if necessary

5$pass = ''; // Change this if necessary

6

7try {

8 $pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass);

9 $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

10} catch (PDOException $e) {

11 echo "Connection failed: " . $e->getMessage();

12}

13?>

**db.php**

php

RunCopy code

1<?php

2include 'config.php';

3

4function getStudents() {

5 global $pdo;

6 $stmt = $pdo->query("SELECT \* FROM Students");

7 return $stmt->fetchAll(PDO::FETCH\_ASSOC);

8}

9

10// Add more functions for other entities as needed

11?>

**student.php**

php

RunCopy code

1<?php

2include 'db.php';

3

4if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

5 $name = $\_POST['name'];

6 $regNo = $\_POST['regNo'];

7 $contactInfo = $\_POST['contactInfo'];

8

9 $stmt = $pdo->prepare("INSERT INTO Students (Name, RegNo, ContactInfo) VALUES (?, ?, ?)");

10 $stmt->execute([$name, $regNo, $contactInfo]);

11 echo "Student registered successfully!";

12}

13?>

**faculty.php**

php

RunCopy code

1<?php

2include 'db.php';

3

4// Similar to student.php, implement faculty functionalities

5?>

**admin.php**

php

RunCopy code

1<?php

2include 'db.php';

3

4// Similar to student.php, implement admin functionalities

5?>

**3. Frontend Code**

**index.html**

html

RunCopy code

1<!DOCTYPE html>

2<html lang="en">

3<head>

4 <meta charset="UTF-8">

5 <meta name="viewport" content="width=device-width, initial-scale=1.0">

6 <link rel="stylesheet" href="style.css">

7 <title>Online Examination System</title>

8</head>

9<body>

10 <h1>Online Examination System</h1>

11 <form action="student.php" method="POST">

12 <h2>Register as Student</h2>

13 <input type="text" name="name" placeholder="Name" required>

14 <input type="text" name="regNo" placeholder="Registration Number" required>

15 <input type="text" name="contactInfo" placeholder="Contact Information" required>

16 <button type="submit">Register</button>

17 </form>

18 <!-- Add forms for Faculty and Admin -->

19</body>

20</html>

**style.css**

css

RunCopy code

1body {

2 font-family: Arial, sans-serif;

3 margin: 20px;

4}

5

6h1, h2 {

7 color: #333;

8}

9