Online Exams Management System (OEMS)
Project Documentation

1. Overview

The **Online Exams Management System** is a web-based application that simplifies exam creation, management, and evaluation. It allows **Administrators** to create and manage courses, exams, and questions, while **Students** can take exams, view their results, and provide feedback.

This system ensures automated exam delivery, instant grading, and efficient management of academic assessments.

2. System Users

2.1 Administrator (Admin)

- Manages courses.
- Creates, updates, and deletes exams.
- Adds, updates, and deletes exam questions.
- Manages student records.
- Views students' feedback.
- Prints/export students' results.

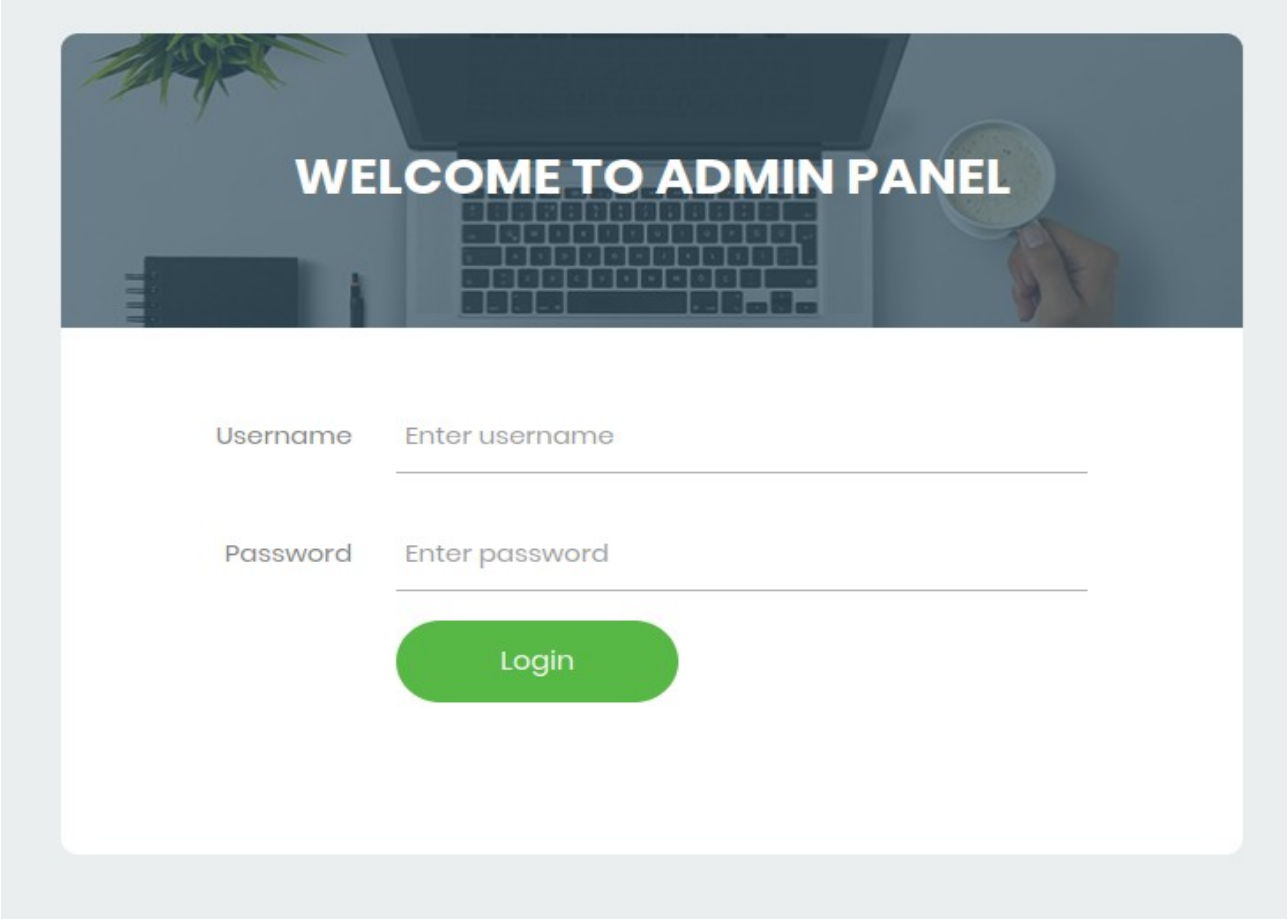
2.2 Student

- Takes exams assigned to their courses.
- Views their exam results.
- Provides feedback after exams.

3. System Features

3.1 Admin Features

- Log in with username and password.

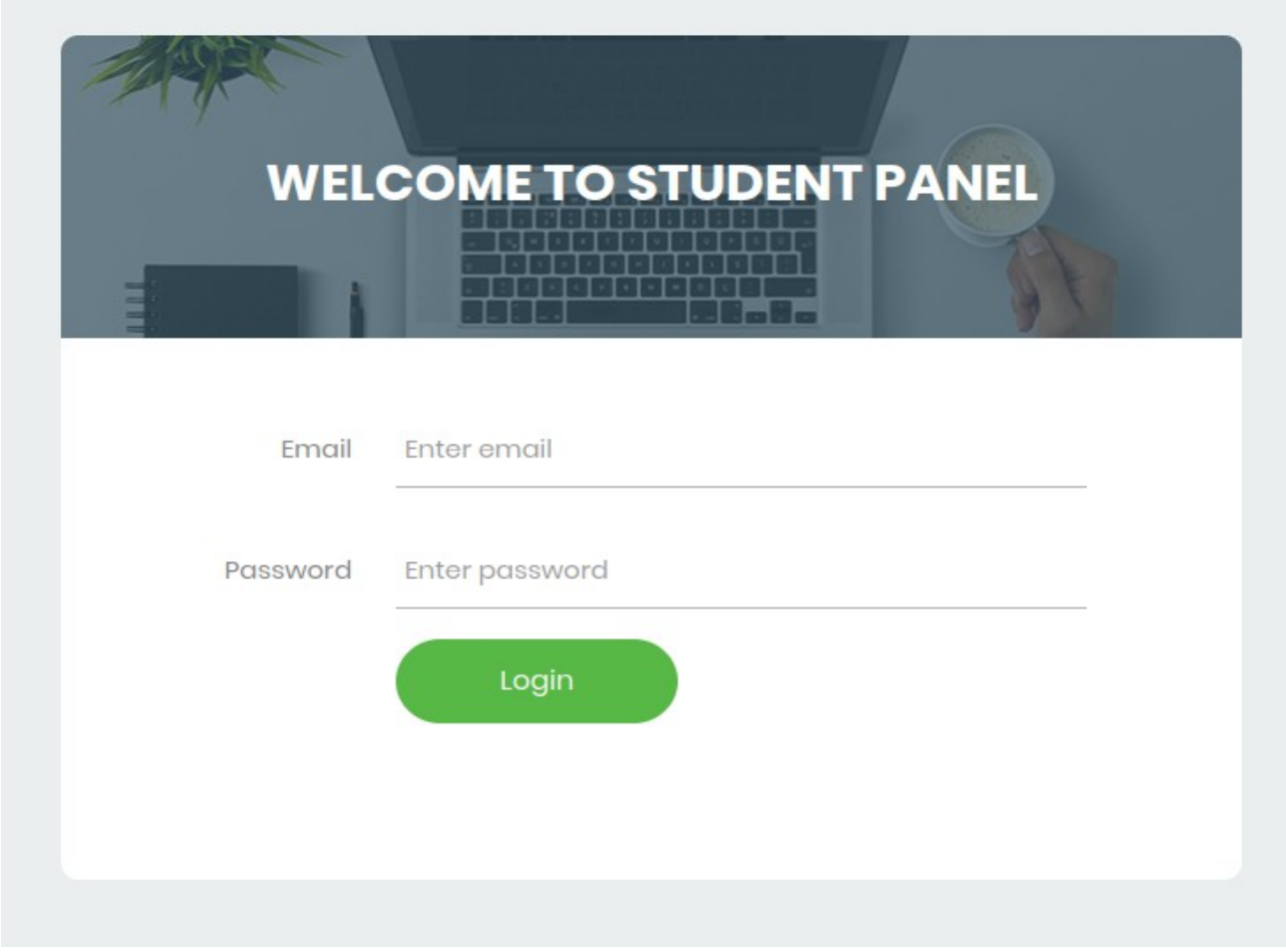


1. **Course Management**
    - Add new courses.
    - Delete existing courses.
2. **Exam Management**
    - Create exams (title, description, start time, duration, course).
    - Update exam details.
    - Delete exams.
3. **Exam Questions Management**
    - Add multiple-choice questions.
    - Update existing questions.
    - Delete questions.

4. **Student Management**
    - Register and manage student profiles.
    - Assign students to courses.
    - Delete/update student details.
5. **Results Management**
    - Generate student scores.
    - Print/export results for records.
6. **Feedback Management**
    - View feedback submitted by students.

3.2 Student Features

1. **Take Exams**
    - Log in with email and password.



- View available exams.
- Attempt exams within the allowed duration.
- Submit answers.

2. **View Results**
    - Access exam results after grading.
    - Check scores per exam.
3. **Provide Feedback**
    - Submit feedback about exams.
    - Feedback is stored for admin review.

4. Database Design

The system is powered by a **relational database** (MySQL). Below is the description of each table from the schema:

4.1 `users` Table

Stores administrator accounts.

- **id**: Primary key.
- **username**: Admin username.
- **password**: Encrypted password.
- **role**: User role (`admin` or `student`).

4.2 `students` Table

Stores student details.

- **id**: Primary key.
- **fullname**: Full name of the student.
- **course_id**: Foreign key → `course.id`.
- **gender**: Student's gender.
- **birthdate**: Date of birth.
- **year_level**: Academic level/year.
- **email**: Student email (unique).
- **password**: Login password (hashed).
- **status**: Active/inactive status.
- **date_registered**: Registration date.

4.3 `course` Table

Stores course details.

- **id**: Primary key.
- **name**: Course name.
- **created_at**: Timestamp of creation.

4.4 `exams` Table

Stores exams information.

- **id**: Primary key.
- **title**: Exam title.
- **start_time**: Scheduled exam start.
- **duration**: Exam duration (minutes).

- **course_id**: Foreign key → `course.id`.
- **description**: Exam description.

4.5 `questions` Table

Stores exam questions.

- **id**: Primary key.
- **exam_question**: Question text.
- **exam_id**: Foreign key → `exams.id`.
- **exam_ch1–exam_ch4**: Multiple-choice options.
- **exam_answer**: Correct answer.
- **exam_status**: Status (active/inactive).

4.6 `exam_questions` Table

Mapping table for exams and questions.

- **id**: Primary key.
- **exam_id**: Foreign key → `exams.id`.
- **question_id**: Foreign key → `questions.id`.

4.7 `exam_attempt` Table

Tracks student exam attempts.

- **examat_id**: Primary key.
- **user_id**: Foreign key → `students.id`.
- **exam_id**: Foreign key → `exams.id`.
- **examat_status**: Status (ongoing, submitted).

- **examat_start_time**: Start time.
- **examat_end_time**: End time.

4.8 `responses` Table

Stores students' answers.

- **id**: Primary key.
- **exam_id**: Foreign key → `exams.id`.
- **user_id**: Foreign key → `students.id`.
- **question_id**: Foreign key → `questions.id`.
- **answer**: Submitted answer.

4.9 `results` Table

Stores exam results.

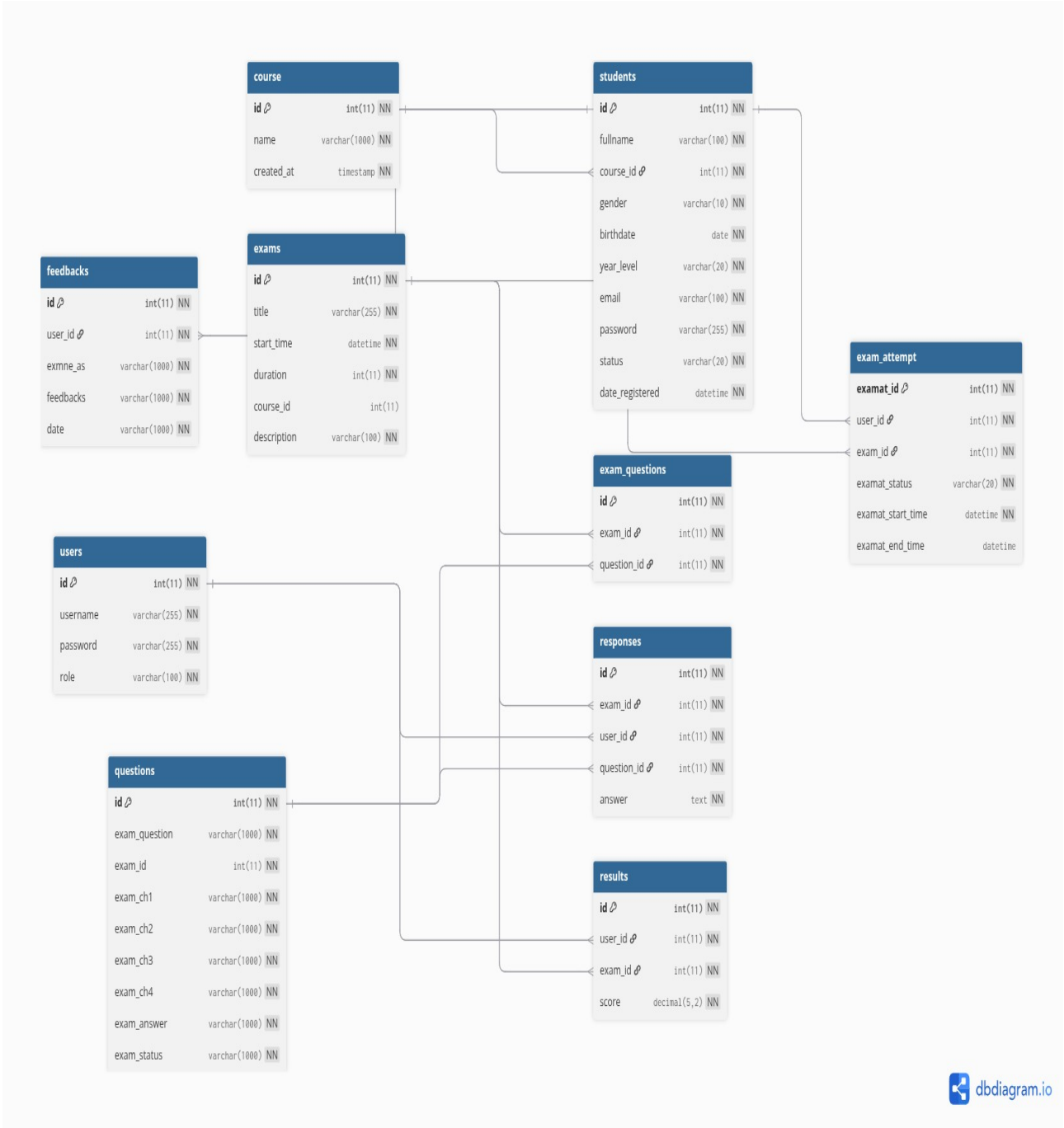
- **id**: Primary key.
- **user_id**: Foreign key → `students.id`.
- **exam_id**: Foreign key → `exams.id`.
- **score**: Student's score.

4.10 `feedbacks` Table

Stores student feedback.

- **id**: Primary key.
- **user_id**: Foreign key → `students.id`.
- **examne_as**: Student's exam alias (if any).
- **feedbacks**: Text feedback.
- **date**: Date submitted.

ENTITY RELATIONSHIP DIAGRAM



5. System Workflow

5.1 Admin Workflow

1. Admin logs in.
2. Admin creates a course.
3. Admin creates an exam under a course.
4. Admin adds questions to the exam.
5. Students attempt exams.
6. System auto-grades answers and saves results.
7. Admin views feedback and prints results.

5.2 Student Workflow

1. Student registers or is registered by admin.
2. Student logs in.
3. Student views available exams.
4. Student attempts the exam within the time limit.
5. Answers are submitted and graded.
6. Student views results.
7. Student submits feedback.

6. Technology Stack

- **Frontend**: HTML, CSS, JavaScript (Bootstrap).
- **Backend**: PHP.
- **Database**: MySQL.
- **Authentication**: Session-based.
- **Reports**: Export results as PDF.

7. Security Features

- Password hashing for users and students.
- Role-based access control (Admin vs Student).
- Prevent multiple submissions of the same exam.
- SQL injection protection via parameterized queries.

8. Future Enhancements

- Add exam proctoring with webcam monitoring.
- Add support for different question types (essay).
- Add ranking system and leaderboards.
- Mobile-friendly responsive UI.