Information Systems and Data Modeling – IT1090

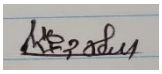
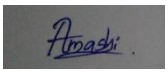
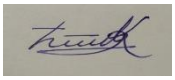
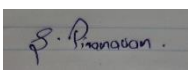
Assignment

Title: Online Teacher Trainer	
Batch Number: 11.1	Group Number: MLB_11.1_09

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged. We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

IT22102782	Afzal Ahamed A H	 …… signature
IT22149626	Chandrasiri G A S D	 …..... signature
IT22150448	Kaushalya W M D S	 …… signature
IT22144430	Liyanage N S	 …… signature
IT22153654	Piranavan S	 …… signature

Submitted on: 04/06/2023

Table of contents

Introduction

An online teacher trainer system is the most recent development in teacher training. Since you may access as many courses as you like and expand your knowledge whenever and wherever you like. Every day, we upgrade our system. Our Edu Teach online teacher trainer is made especially for trainers and students who are eager to acquire new material and gain better management skills for cutting-edge information technology tools. There is also a customer support system that is available 24/7. We ensure that every component is understood by our users as the internationally revised online teacher trainer system.

It also helps with instruction at esteemed universities. It helps educators accomplish their objectives. Your strain of finding reliable research is lessened by our new expert. A variety of technologies must be explored and understood to develop an online teacher trainer website. Programming languages (such as HTML, CSS, and JavaScript) and relational databases were among them (Such as My SQL).

A database must be able to manage a wide range of data. A database is required because managing a computer-based stored data system is easier than managing written or typed documents. Customers and administrators save time and effort because of this. Furthermore, since the data is stored in a database, there is no risk of losing it. The system's users are the only ones who can control it. Collecting requirements and doing requirement analysis while building a database is vital The functional and nonfunctional specifications, as well as the data requirements, can then be determined.

Hypothetical Scenario

A web-based learning management system (LMS) is named the Online Trainer System (OTS). This system caters to two main user groups: Trainee Teachers and Lecturers. Both user groups have access to view the system's content.

When an unregistered user visits the system, they are presented with a form that they can fill out. In this scenario, the form is then reviewed by an administrator who assesses the information provided. If the administrator finds the form suitable, the unregistered user is then granted registered user status. The registered user can be either a Trainee Teacher or a Lecturer.

In order to participate in courses, attend lectures, and take tests, Trainee Teachers are required to make payments for enrollment. Upon successful completion of the course and passing the exams, Trainee Teachers are awarded certificates to acknowledge their accomplishment. On the other hand, Lecturers are responsible for delivering lectures, adding lecture materials to the system, and are compensated for their services. The course structure consists of multiple modules, each of which includes lectures and several exams.

In this hypothetical scenario, the system's development, security, error handling, and updates are overseen by an administrator and coordinator. They ensure that the system functions smoothly and remains up-to-date.

Requirements Analysis

Functional requirements

Functional requirements mean how registered users interact with the system. Online teacher training system also includes a set of main Functional requirements**.** There are as follows.
The website can be accessed by System Administrator, Registered Lectures, and Registered training teachers.

System Administrator –

- System Administrator has access to the both lectures side and training teacher side of the system
- System Administrator can view and manage data.
- System Administrator can edit the database.
- System Administrator can manage registered user profiles.
- System Administrator is responsible for maintaining the security of the system
- System Administrator is responsible for verifying the privacy of registered users.

Lecture –

- Lecture has access to the lectures side of the system.
- Lecture can manage a learning environment to train teachers.
- Lecture can update lecture notes to training teachers.
- Lecture can plan learning schedules for training teachers.
- Lecture can upload online tests to training teachers.
- Lecture can mark the answer sheets of training teachers.
- Lecture can upload marking scheme and result sheet.

Training Teacher –

- Training Teacher has access to the training teacher side of the system.
- Training Teacher has access to the system.
- Training Teacher can view learning schedules.
- Training Teacher can download the lecture notes.
- Training Teacher can answer online tests.
- Training Teacher can upload the answer sheets.
- Training Teacher can download the marking scheme and Result sheet.

Nonfunctional requirements

Non-functional requirements are a set of specifications that describe the operational capabilities of the system. And it further means that NFRS can be defined as a quality attribute. Functional requirements as well as non-functional requirements are important for the system and its functionality.

Availability

- The system should be available 24 hours for users.

Speed

The website's load time should not be more than 03 seconds for users.

Security

- The system's information should be accessible to only registered users.
- The system's special information should not be accessible to unregistered users.
- Data of the system should be stored in a protective method.
- The privacy of the registered users should be verified.

Scalability

- More than 2000 registered users should have accessible at a time.

Usability

- The system should be simple and understandable to the registered users.

Data Requirements

Unregistered User

- NIC
- Name (First name and Last name)
- Email
- Password
- Mobile No

Registered User

- Reg_no
- Name (First name and Last name)
- Email
- Password
- Mobile No

Trainee Teacher

- T_No
- Name (First name and Last name)
- Email
- Password
- Mobile No

Lecturer

- L_No
- Name (First name and Last name)
- Email
- Password
- Mobile No

Course

- C_ID
- Name
- Duration

Module

- M_ID
- Name
- Credit

Lecture

- Lec_No
- Name
- Type
- Duration

Exam

- E_no
- Name
- Duration

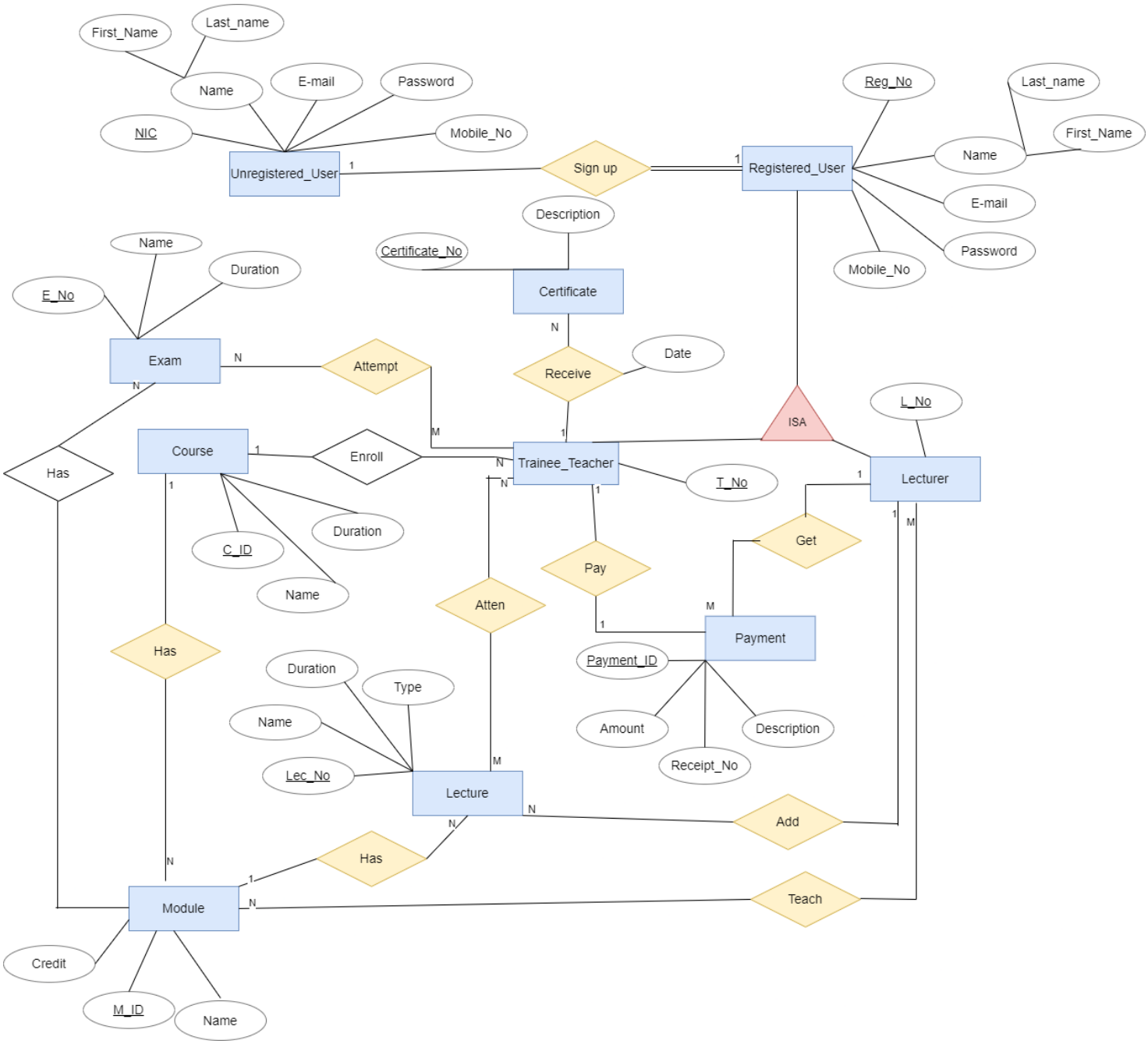
Payment

- Payment_ID
- Amount
- Receipt No
- Description

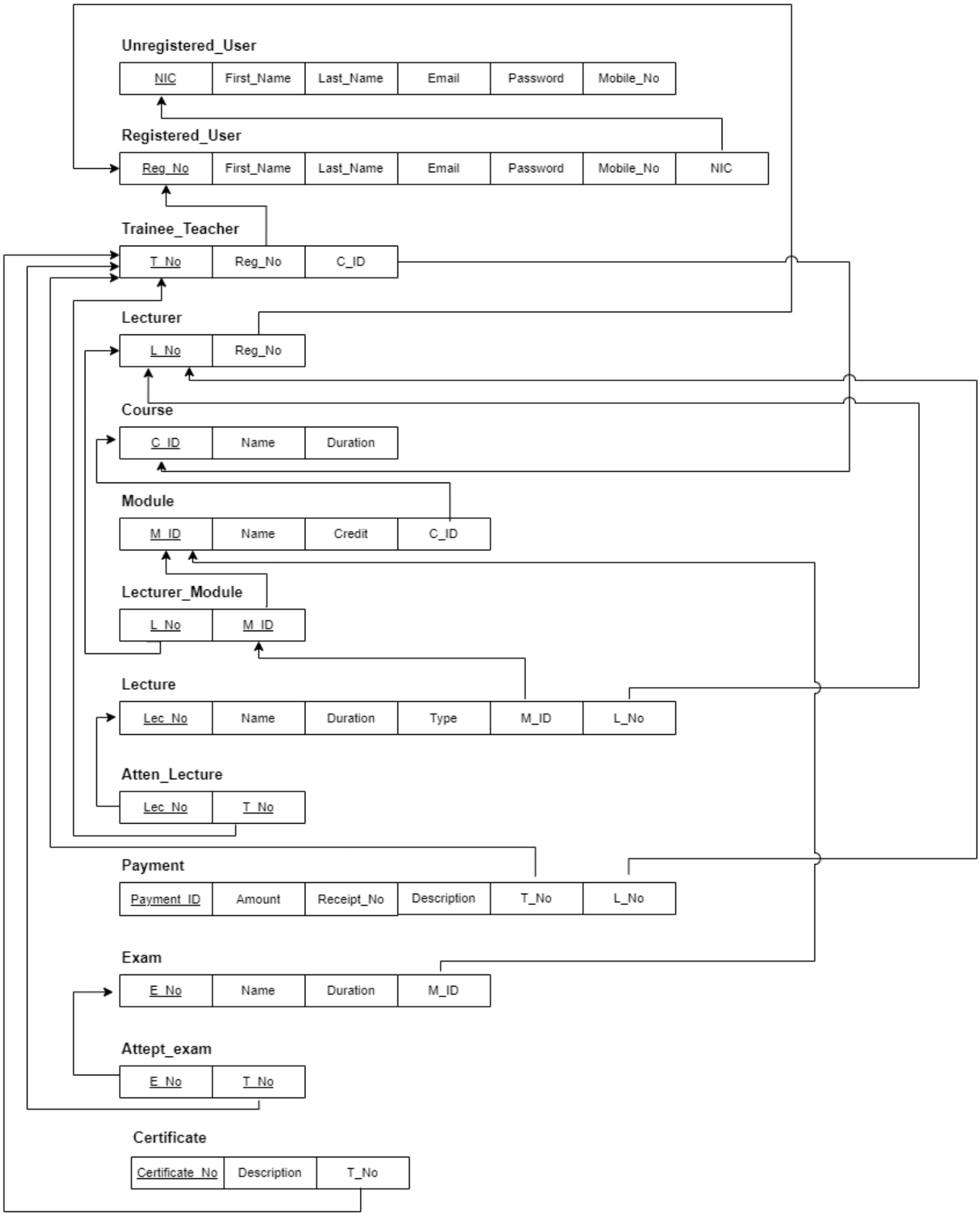
Certificate

- Certificate_No
- Description

Entity Relation Diagram



Relational Schema



```sql
--Unregister User Table--
CREATE TABLE Unregister_User(
	NIC VARCHAR(12),
	First_Name VARCHAR(50),
	Last_Name VARCHAR(50),
	Email VARCHAR(50),
	Password VARCHAR(20),
	Mobile_NO INT,
	constraint PK_UU primary key(NIC)
);

--Register User Table--
CREATE TABLE Register_User(
	Reg_No CHAR(5),
	First_Name VARCHAR(50),
	Last_Name VARCHAR(50),
	Email VARCHAR(50),
	Password VARCHAR,
	Mobile_NO INT,
	NIC VARCHAR(12),
	constraint PK_RU primary key(Reg_No),
	constraint FK_RU foreign key(NIC) references Unregister_User(NIC),
	constraint Register_User_Reg_No_CK check (Reg_No like '[r\R][0-9][0-9][0-9][0-9]')
);

--Course Table--
CREATE TABLE Course(
	C_ID CHAR(5),
	Name VARCHAR (5),
	Duration VARCHAR(30),
	constraint PK_C primary key(C_ID),
	constraint Course_C_No_CK check (C_ID like '[t\T][0-9][0-9][0-9][0-9]')
);


--Trainee Teacher Table--
CREATE TABLE Trainee_Teacher(
	T_No CHAR(5),
	Reg_No CHAR (5),
	C_ID CHAR(5),
	constraint PK_TT primary key(T_No),
	constraint FK_TT foreign key(Reg_No) references Register_User(Reg_No),
	constraint Trainee_Teacher_T_No_CK check (T_No like '[t\T][0-9][0-9][0-9][0-9]'),
	constraint Trainee_Teacher_Reg_No_CK check (Reg_No like '[r\R][0-9][0-9][0-9][0-9]'),
	constraint Trainee_Teacher_C_ID_CK check (C_ID like '[c\C][0-9][0-9][0-9][0-9]')
);

--Lecturer Table--
CREATE TABLE Lecturer(
	L_No CHAR(5),
	Reg_No CHAR (5),
	constraint PK_L primary key(L_No),
	constraint FK_L foreign key(Reg_No) references Register_User(Reg_No),
	constraint Lecturer_L_No_CK check (L_No like '[1\L][0-9][0-9][0-9][0-9]'),
	constraint Lecturer_Reg_No_CK check (Reg_No like '[r\R][0-9][0-9][0-9][0-9]')
);
--Modules Table--
CREATE TABLE Modules(
	M_ID CHAR(5),
	Name VARCHAR(50),
	Credit int,
	C_ID CHAR(5),
	constraint PK_M primary key(M_ID),
	constraint FK_M foreign key(C_ID) references Course(C_ID),
	constraint Modules_M_No_CK check (M_ID like '[m\M][0-9][0-9][0-9][0-9]'),
	constraint Modules_C_ID_CK check (C_ID like '[r\R][0-9][0-9][0-9][0-9]')
);
```

```
--Lecturer_Module Table--
CREATE TABLE Lecturer_Module(
	L_No CHAR(5),
	M_ID CHAR(5),
	constraint PK_LM primary key(L_No,M_ID),
	constraint FK_LM1 foreign key(L_No) references Lecturer(L_No),
	constraint FK_LM2 foreign key(M_ID) references Modules(M_ID),
	constraint Lecturer_Module_L_No_CK check (L_No like '[l\L][0-9][0-9][0-9][0-9]'),
	constraint Lecturer_Module_M_ID_CK check (M_ID like '[m\M][0-9][0-9][0-9][0-9]')
);

--Lecture Table--
CREATE TABLE Lecture(
	Lec_No CHAR(5),
	Name VARCHAR(50),
	Duration VARCHAR(50),
	Type VARCHAR(50),
	M_ID char(5),
	L_No char(5),
	constraint PK_Lec primary key(Lec_No),
	constraint FK_Lec1 foreign key(M_ID) references Modules(M_ID),
	constraint FK_Lec2 foreign key(L_No) references Lecturer(L_No),
	constraint Lecture_Lec_No_CK check (Lec_No like '[LEC\Lec\lec][0-9][0-9][0-9][0-9]'),
	constraint Lecture_M_ID_CK check (M_ID like '[m\M][0-9][0-9][0-9][0-9]'),
);
--Atten_Lecture Table--
CREATE TABLE Atten_Lecture(
	Lec_No CHAR(5),
	T_No CHAR (5),
	constraint PK_AL primary key(Lec_No,T_No),
	constraint FK_AL1 foreign key(Lec_No) references Lecture(Lec_No),
	constraint FK_AL2 foreign key(T_No) references Trainee_Teacher(T_No),
	constraint Atten_Lecture_Lec_No_CK check (Lec_No like '[l\L][0-9][0-9][0-9][0-9]'),
	constraint Atten_Lecture_T_No_CK check (T_No like '[t\T][0-9][0-9][0-9][0-9]')
);

--Payment Table--
CREATE TABLE Payment(
	Payment_ID CHAR(5),
	Amount FLOAT,
	Receipt_No VARCHAR(50),
	Description VARCHAR(100),
	T_NO CHAR (5),
	L_NO CHAR (5),
	constraint PK_P primary key(Payment_ID),
	constraint FK_P1 foreign key(T_NO) references Trainee_Teacher(T_NO),
	constraint FK_P2 foreign key(L_NO) references Lecturer(L_NO),
	constraint Payment_Payment_ID_CK check (Payment_ID like '[p\P][0-9][0-9][0-9][0-9]'),
	constraint Payment_T_NO_CK check (T_NO like '[t\T][0-9][0-9][0-9][0-9]'),
	constraint Payment_L_NO_CK check (L_NO like '[l\L][0-9][0-9][0-9][0-9]')
);
--Exam Table--
CREATE TABLE Exam(
	E_No CHAR(5),
	Name VARCHAR(50),
	Duration VARCHAR(50),
	M_ID CHAR(5),
	constraint PK_E primary key(E_No),
	constraint FK_E foreign key(M_ID) references Modules(M_ID),
	constraint Exam_E_No_CK check (E_No like '[e\E][0-9][0-9][0-9][0-9]'),
	constraint Exam_M_ID_CK check (M_ID like '[M\m][0-9][0-9][0-9][0-9]')
);
```

```
--Attept_Exam Table--
CREATE TABLE Attept_Exam(
	E_No CHAR(5),
	T_No CHAR (5),
	constraint PK_AE primary key(E_No,T_No),
	constraint FK_AE1 foreign key(E_No) references Exam(E_No),
	constraint FK_AE2 foreign key(T_No) references Trainee_Teacher(T_No),
	constraint Attept_Exam_E_No_CK check (E_No like '[e\E][0-9][0-9][0-9][0-9]'),
	constraint Attept_Exam_T_No_CK check (T_No like '[t\T][0-9][0-9][0-9][0-9]')
);
--Certificate Table--
CREATE TABLE Certificate(
	Certificate_No CHAR(5),
	Description VARCHAR(100),
	T_No CHAR (5),
	constraint PK_Certificate primary key(Certificate_No),
	constraint FK_Certificate foreign key(T_No) references Trainee_Teacher(T_No),
	constraint Certificate_T_No_CK check (T_No like '[r\R][0-9][0-9][0-9][0-9]')
);


-- INSERTION OF SAMPLE DATA –

--Unregister_User Table--
insert into Unregister_User values(20021661045,'John
','Smith','johnsmith@example.com','Js12345',0765493157);
insert into Unregister_User values(20021666045,'Emma
','Johnson','emmajohnson@example.com','Ej56789',0795468541);
insert into Unregister_User values(20021661070,'Michael
','Williams','michaelwilliams@example.com','Mw98765',0725486148);
insert into Unregister_User values(20021630045,'Sophia
','Brown','sophiabrown@example.com','Sb43210',0715489354);
insert into Unregister_User values(20021881045,'Daniel
','Davis','danieldavis@example.com','Dd24680',0754863249);
insert into Unregister_User values(20034661045,'Olivia
','Taylor','oliviataylor@example.com','Ot13579',0763548924);
insert into Unregister_User
values(20015666045,'David','Martinez','davidmartinez@example.com','Dm02468',0741239586);
insert into Unregister_User values(20074461070,'Isabella
','Anderson','isabellaanderson@example.com','Ia75319',0759843258);
insert into Unregister_User
values(20025000045,'James','Wilson','jameswilson@example.com','Jw95162',0742158936);
insert into Unregister_User
values(20021121045,'Emily','Thompson','emilythompson@example.com','Et86420',0778965423);

--Register_User Table--
insert into Register_User values('R0001','John
','Smith','johnsmith@example.com','Js12345',0765493157,20021661045);
insert into Register_User values('R0002','Emma
','Johnson','emmajohnson@example.com','Ej56789',0795468541,20021666045);
insert into Register_User values('R0003','Michael
','Williams','michaelwilliams@example.com','Mw98765',0725486148,20021661070);
insert into Register_User values('R0004','Sophia
','Brown','sophiabrown@example.com','Sb43210',0715489354,20021630045);
insert into Register_User values('R0005','Daniel
','Davis','danieldavis@example.com','Dd24680',0754863249,20021881045);
insert into Register_User values('R0006','Olivia
','Taylor','oliviataylor@example.com','Ot13579',0763548924,20034661045);
insert into Register_User
values('R0007','David','Martinez','davidmartinez@example.com','Dm02468',0741239586,20015666045);
insert into Register_User values('R0008','Isabella
','Anderson','isabellaanderson@example.com','Ia75319',0759843258,20074461070);
insert into Register_User
values('R0009','James','Wilson','jameswilson@example.com','Jw95162',0742158936,20025000045);
insert into Register_User
values('R0010','Emily','Thompson','emilythompson@example.com','Et86420',0778965423,20021121045);
```

```
--Course Table--
insert into Course values ('C0001', 'Pedagogy and Classroom Management', '10 months');
insert into Course values ('C0002', 'Technology in Education', '12 months');
insert into Course values ('C0003', 'Differentiated Instruction', '5 months');
insert into Course values ('C0004', 'Special Education', '11 months');
insert into Course values ('C0005', 'Culturally Responsive Teaching', '6 months');
--Trainee_Teacher Table--
insert into Trainee_Teacher values('T0001','R0001','C0002');
insert into Trainee_Teacher values('T0002','R0002','C0001');
insert into Trainee_Teacher values('T0003','R0003','C0005');
insert into Trainee_Teacher values('T0004','R0004','C0003');
insert into Trainee_Teacher values('T0005','R0005','C0004');

--Lecturer Table--
insert into Lecturer values('L0001','R0006');
insert into Lecturer values('L0002','R0007');
insert into Lecturer values('L0003','R0008');
insert into Lecturer values('L0004','R0009');
insert into Lecturer values('L0005','R0010');

--Modules Table--
insert into Modules values('M0001', 'Introduction to Pedagogy', 2, 'C0001');
insert into Modules values('M0002', 'Digital Literacy and Citizenship', 4, 'C0002');
insert into Modules values('M0003', 'Understanding Learning Styles', 3, 'C0003');
insert into Modules values('M0004', 'Laws and Regulations in Special Education', 4, 'C0004');
insert into Modules values('M0005', 'Anti-Bias Education', 2, 'C0005');

--Lecturer_Module Table--
insert into Lecturer_Module values('L0001','M0001');
insert into Lecturer_Module values('L0002','M0002');
insert into Lecturer_Module values('L0003','M0003');
insert into Lecturer_Module values('L0004','M0004');
insert into Lecturer_Module values('L0005','M0005');

--Lecture Table--
insert into Lecture values('LEC0001', 'Introduction to Pedagogy', '1 hour', 'Classroom',
'M0001', 'L0001');
insert into Lecture values('LEC0002', 'Introduction to Educational Technology', '1 hour',
'Classroom', 'M0002', 'L0002');
insert into Lecture values('LEC0003', 'Introduction to Differentiated Instructionn', '1 hour',
'Classroom', 'M0003', 'L0003');
insert into Lecture values('LEC0004', 'Introduction to Special Education', '1 hour',
'Classroom', 'M0004', 'L0004');
insert into Lecture values('LEC0005', 'Introduction to Culturally Responsive Teaching', '1
hour', 'Classroom', 'M0005', 'L0005');


--Atten_Lecture Table--
insert into Atten_Lecture values ('LEC0001', 'T0001');
insert into Atten_Lecture values ('LEC0002', 'T0002');
insert into Atten_Lecture values ('LEC0003', 'T0003');
insert into Atten_Lecture values ('LEC0004', 'T0004');
insert into Atten_Lecture values ('LEC0005', 'T0005');


--Payment Table--
insert into Payment values  ('P0001', 50000.00, 'RCPT001', 'Course fee', 'T0001', NULL);
insert into Payment values  ('P0002', 60000.00, 'RCPT001', 'Salary', NULL, 'L0001');
insert into Payment values  ('P0003', 40000.00, 'RCPT001', 'Course fee', 'T0002', NULL);
insert into Payment values  ('P0004', 80000.00, 'RCPT001', 'Salary', NULL, 'L0002');
insert into Payment values  ('P0005', 30000.00, 'RCPT001', 'Course fee', 'T0003', NULL);
```

```sql
--Exam Table--
insert into Exam values ('E0001', 'Final Exam', '2 hours', 'M0001');
insert into Exam values ('E0002', 'Mid Exam', '2 hours', 'M0002');
insert into Exam values ('E0003', 'Final Exam', '2 hours', 'M0003');
insert into Exam values ('E0004', 'Mid Exam', '2 hours', 'M0004');
insert into Exam values ('E0005', 'Final Exam', '2 hours', 'M0005');

--Attept_Exam Table--
insert into Attept_Exam values  ('E0001', 'T0001');
insert into Attept_Exam values  ('E0001', 'T0002');
insert into Attept_Exam values  ('E0003', 'T0001');
insert into Attept_Exam values  ('E0003', 'T0002');
insert into Attept_Exam values  ('E0005', 'T0003');

--Certificate Table--
insert into Certificate values  ('00001', 'Certificate of Completion', 'T0001');
insert into Certificate values  ('00002', 'Certificate of Completion', 'T0002');
insert into Certificate values  ('00003', 'Certificate of Completion', 'T0003');
insert into Certificate values  ('00004', 'Certificate of Completion', 'T0004');
insert into Certificate values  ('00005', 'Certificate of Completion', 'T0005');
```

Special Performance Considerations

- **Scalability:**
  - Design for handling a growing number of users and training sessions.
  - Employ techniques like load balancing and horizontal scaling.
- **Response Time:**
  - Minimize latency issues for real-time interactions.
  - Optimize network communication and processing time.
- **Bandwidth Optimization:**
  - Implement adaptive streaming to adjust video quality based on the user's internet connection.
  - Reduce buffering for smoother playback.
- **Compatibility and Accessibility:**
  - Ensure compatibility across web browsers and operating systems.
  - Adhere to accessibility standards for users with disabilities.
- **Data Storage and Retrieval:**
  - Design efficient mechanisms for storing and retrieving user profiles, training materials, and progress records.
  - Utilize database indexing, caching, and data compression techniques.
- **Security:**
  - Implement robust security measures to protect sensitive information.
  - Include encryption, user authentication, secure transmission protocols, and regular security audits.
- **System Monitoring and Analytics:**
  - Set up monitoring tools to track system performance and identify bottlenecks.
  - Utilize real-time analytics for insights into user behavior and performance metrics.
- **Usability and User Interface Design:**
  - Create an intuitive and user-friendly interface for easy navigation.
  - Optimize the user interface to minimize cognitive load and provide clear instructions.

Special Security Requirements

User Authentication and Access Control

- Strong password policies.
- Two-factor authentication or biometric authentication.
- Role-based access controls.

Secure Data Transmission

- Encryption using HTTPS (SSL/TLS).

Data Encryption

- Encryption of sensitive data at rest and in transit.

Robust Authorization Mechanism

- Well-defined access restrictions and permissions.

Secure Infrastructure

- Secure servers and networks.
- Regular application of security patches and updates.

Secure Coding Practices

- Adherence to secure coding standards.
- Regular security code reviews and testing.

Regular Data Backups

- Automated and secure data backup procedures.

Logging and Monitoring

- Comprehensive logging of user activities.
- Monitoring of system logs, network traffic, and user behavior.

Secure Third-Party Integrations

- Validation of security practices of external services or APIs.
- Secure API authentication and data encryption.

Staff Training and Awareness

- Security best practices education for administrators, developers, and users.

Incident Response and Recovery

- Incident response plan.
- Reporting and responding to security breaches.
- System restoration after an incident.