**A close up of a sign

Description automatically generatedAMERICAN INTERNATIONAL**

**UNIVERSITY-BANGLADESH**

**Faculty of Science and Technology**

**Project Report (Mid)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title: | Online Exam Management System | | | |
| Group Name: | Gliders | | Date of Submission: | 24 March 2022 |
| Course Title: | Advance Database Management System | | | |
| Course Code: | CSC4181 | | Section: | C |
| Semester: | Spring | 2022-23 | Course Teacher: | Rezwan Ahmed |

**Declaration and Statement of Authorship:**

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student’s work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

*\* Student(s) must complete all details except the faculty use part.*

\* Please submit all assignments to your course teacher or the office of the concerned teacher.

|  |  |
| --- | --- |
| Group Name/No.: Gliders |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **ID** | **Program** | **Signature** |
| 1 | Rahat, Md Mohibor Rahman | 19-39517-1 | BSc [CSE] |  |
| 2 | Moni, Khuko | 19-39501-1 | BSc [CSE] |  |

|  |  |  |
| --- | --- | --- |
| ***Faculty use only*** | | |
| FACULTYCOMMENTS | **Marks Obtained** |  |
|  |
|  |
|  | **Total Marks** |  |
|  |

**Table of Contents**

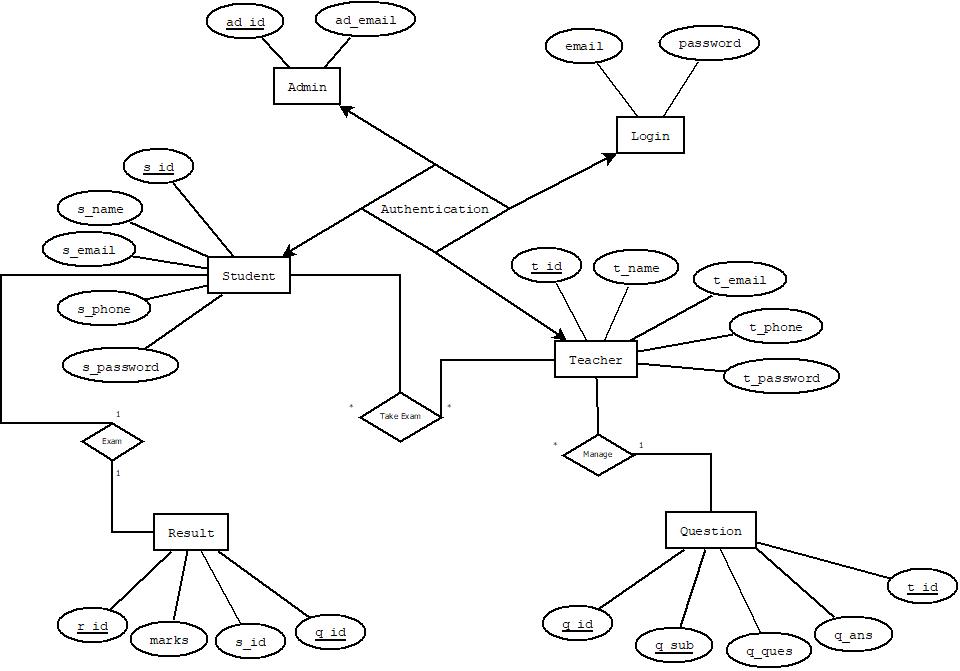
|  |  |  |
| --- | --- | --- |
| No. | Topic | Page No |
| 01 | **System Summary** | 03 |
| 02 | **ERD Diagram** | 04 |
| 03 | **Class Diagram of the System** | 05 |
| 04 | **Use Case Diagrams of the System** | 06 |
| 05 | **Activity Diagrams of the System** | 07 |
| 06 | **Database Schema Diagram** | 08 |
| 07 | **Sample Data** | 09-10 |
| 08 | **Questions for Query and Solve** | 11-12 |
| 09 | **UI Design (Login and Registration Functionality Implementation)** | 13-18 |
| 10 | **Database (SQL)** | 19-24 |

**System Summary**

This System is about an Online Exam Management System. Where there are mainly three types of users (Admin, Teacher & Student). In this system these users will be able to manage all the events regrading online examinations. The main two users (Teachers and Students) will be able to register to the system using a registration form but they won’t be able to login to the system unless an admin approves the regarding user. After getting the approval from an admin the users will be able to login as the user type they were permitted to. After login the teachers will be able to schedule an exam, create/delete/edit questions and answers and also check results. The student users will be able to give the exam and also check their marks after finishing the exam. Admins will have the authority to check marks/ schedules/user information and also approve or delete any user. This whole project is online based.

**ERD Diagram of the System**

***Scenario:***

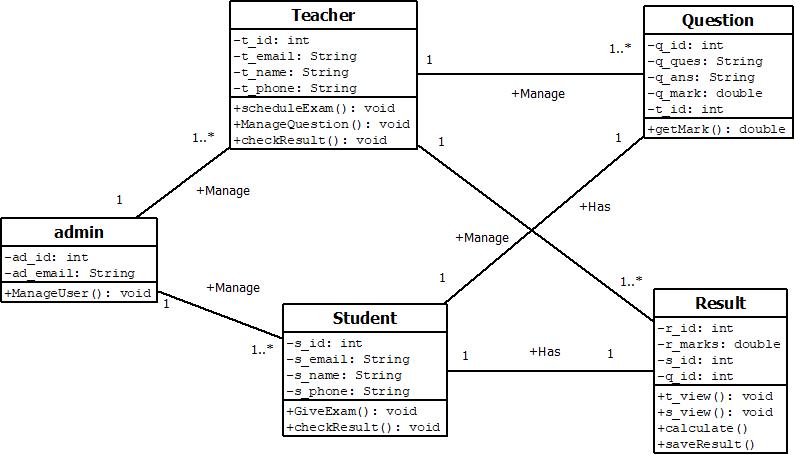
1. Each admin has a unique admin id.
2. Each teacher has a unique teacher id, teacher name, email, and phone number.
3. Each student has a unique student id, student name, email, and phone number.
4. Teacher manages subject & takes exam of students. And student takes subject. Each Subject has a name and unique subject id. A student can take more than one subjects.
5. A question can have many answers. Question is identified by question id & it has question value also.
6. Teacher takes exams & students take part in exam. Different exam happens for different subjects. For each student, exam has different results. Each result has unique string id and value.

***Fig:*** ERD Diagram

**Class Diagram of the System**

***Scenario:***

There are five classes in our system which are Admin, Student, Teacher, Question and Result.

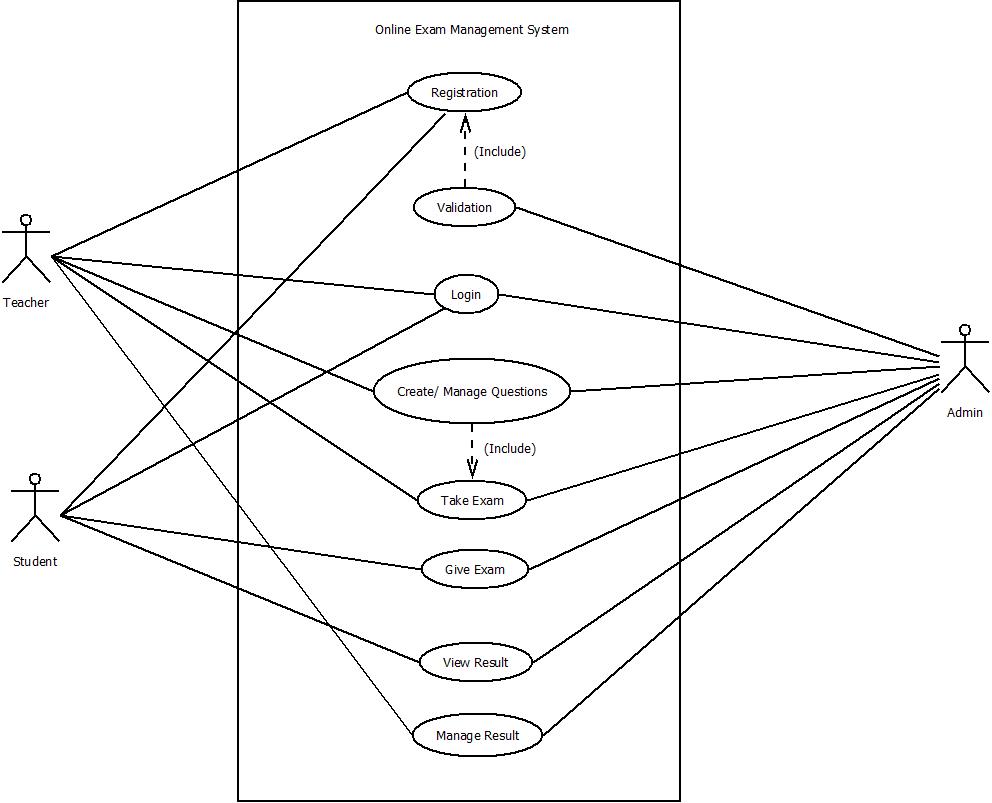
1. Admin has their id which is integer and their name which is string both of them are private. In admin class admin id is the primary key. In the admin class there are only one operation which is manage user which return type is void. Admins can approve and modify users if necessary through this function.
2. Student class has their own student id, email, and name and phone number. A student has two operations which are given exam, and check result.
3. Teacher class has their own teacher id, email, and name and phone number. A teacher has three operations which are schedule exam, manage question and manage result.
4. Question class has its own question id, questions, answers and marks related to the questions. It has only one operation to return marks which return type is double.
5. Result class has result id, marks and cgpa where there are four functions for teacher and student separate result view, calculate result and save result.

***Fig:*** Class Diagram

**Use Case Diagram of the System**

***Scenario:***

This is a use case diagram for Online Exam Management System. Here we have three types of actors (Teacher, Student and Admin). Users will complete their registration first. Then registration will be checked or verified by the admin. After verification the respected user will be able to login to the system. After login user (student) can take exam and see the results from their previous exams. Same process will be for the other user (teacher). The other user will be able to create or mange the questions and also view results for all the relative subjects.

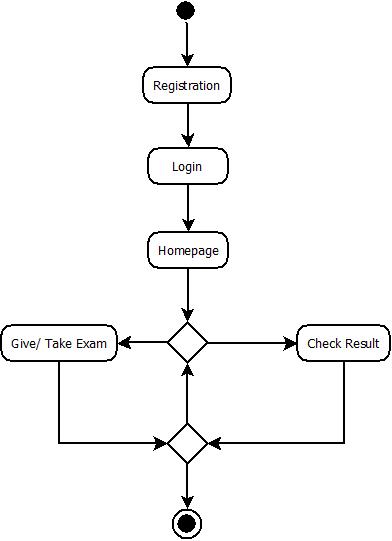
******

***Fig:*** Use Case Diagram

**Activity Diagrams of the System**

***Scenario:***

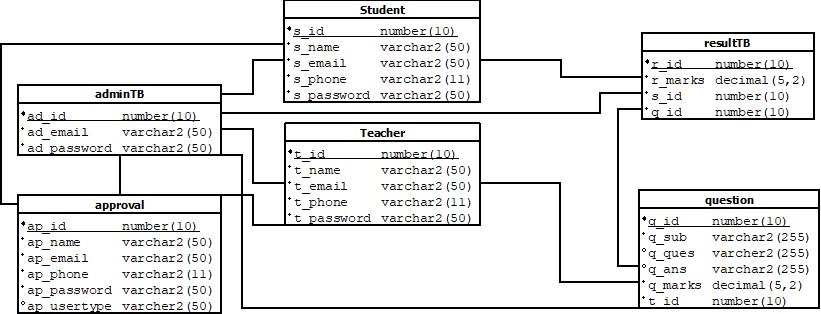
After starting the process user need to register first. Then they will login and enter to the home page. After coming home page user (student) can take exam and see the results from their previous exams. Same process will be for the other user (teacher). The other user will be able to create or mange the questions and also view results for all the relative subjects. After all this the process will end.

***Fig:*** Activity Diagram

**Database Schema Diagram**

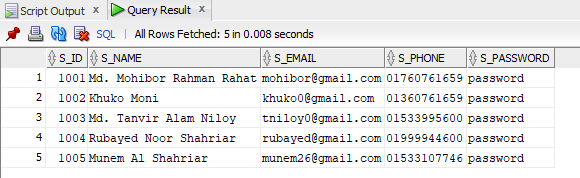
***Scenario:***

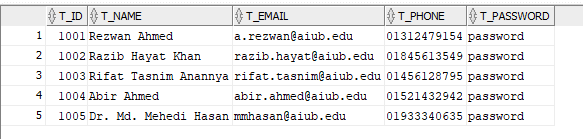
In the schema diagram there are 6 table with different entity and their attribute. They are teacher, student, adminTB, approval, question and resultTB. Attributes of the admin table are ad\_id, ad\_email and ad\_password. Here ad\_id is the primary key. Attributes of the student table are s\_id, s\_name, s\_email, s\_phone and s\_password. Here s\_id is the primary key. Attributes of the teacher table are t\_id, t\_name, t\_email, t\_phone and t\_password. Here t\_id is the primary key. Attributes of the question table are q\_id, q\_sub, q\_ques, q\_ans and t\_id. Here q\_id is the primary key and t\_id is the foreign key from teacher table. Attributes of the resultTB table are r\_id, r\_marks, q\_id and s\_id. Here r\_id is the primary key and q\_id and s\_id are the foreign keys from question table and student table respectively. The attributes of the approval table are ap\_id, ap\_name, ap\_email, ap\_phone, ap\_password and ap\_usertype. We connected the tables regarding their primary key and foreign key.

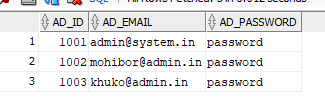
**

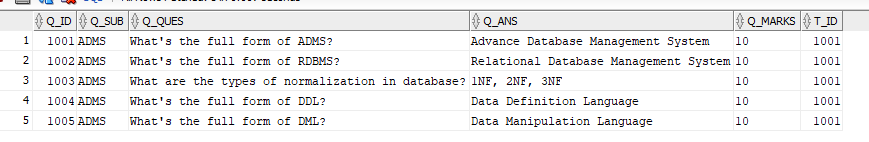
***Fig:*** Activity Diagram

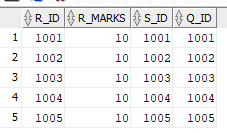
**Sample Data**

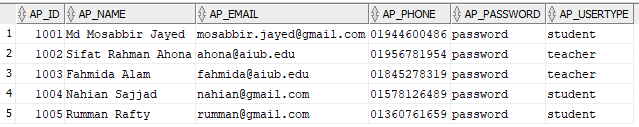
***Student Table:***

***Teacher Table:***

 ***Admin Table:***

***Question Table:***

***Result Table:***

***Approval Table:***

**Questions for Query and Solve**

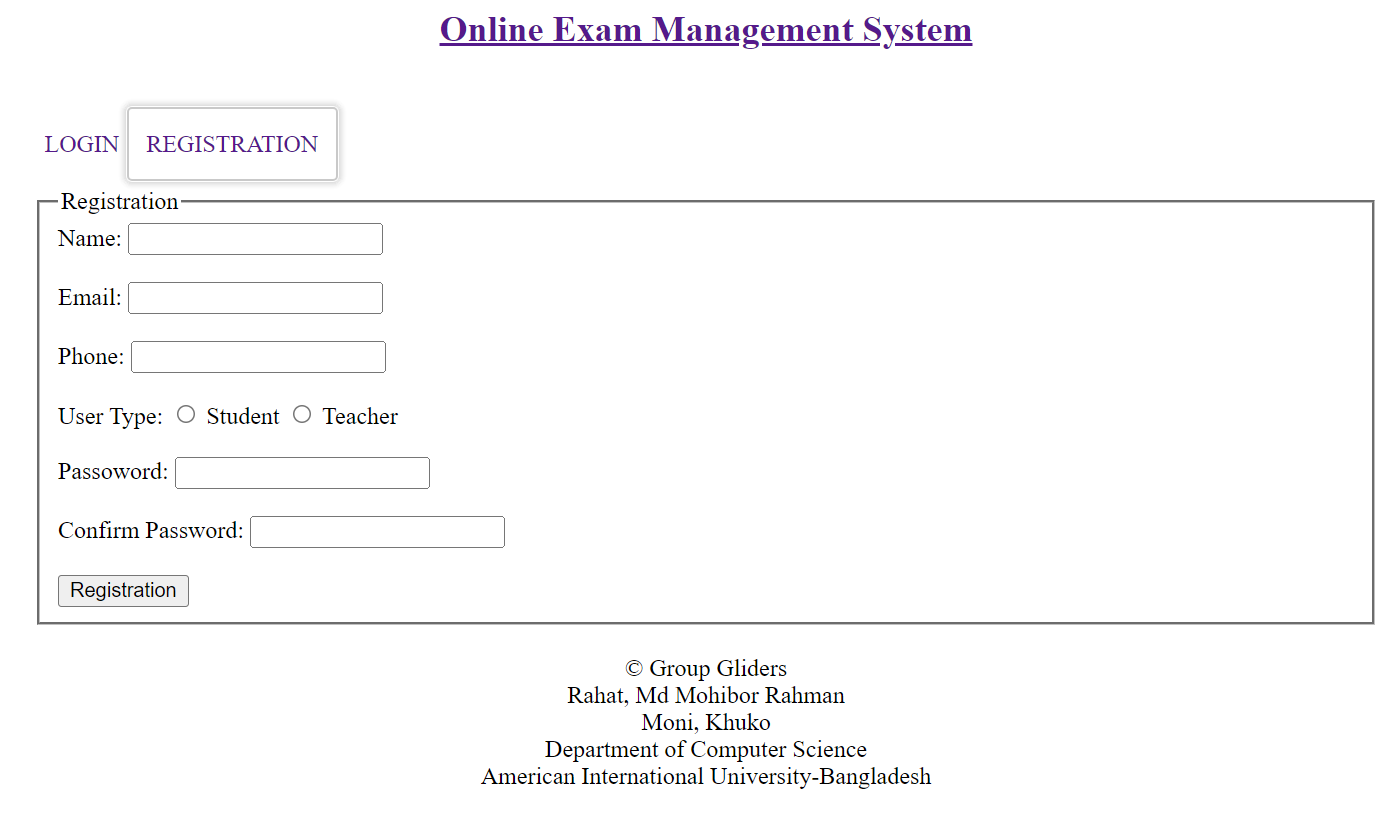
***Questions:***

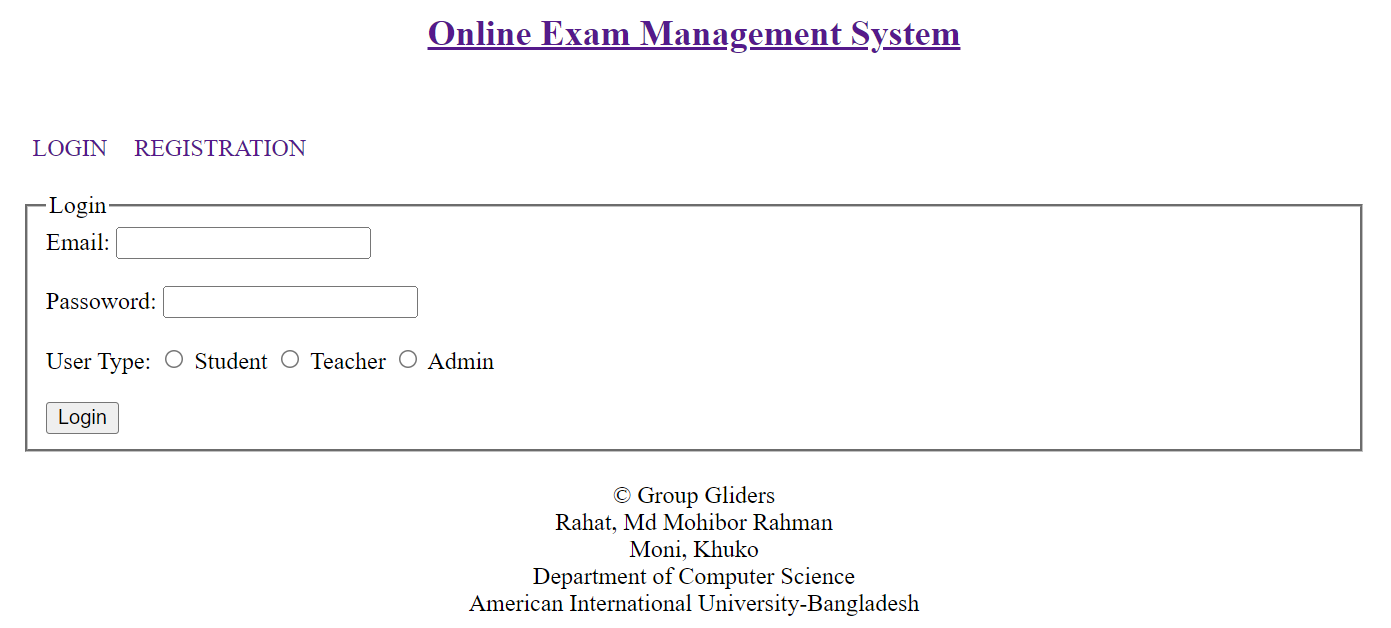
1. Display all the students name and phone number only.
2. Make teacher email unique.
3. Make student phone unique.
4. Show all the results obtained by a student.
5. Find Number of teachers whose first name is ‘Rezwan’.
6. Write a query using join to find the teacher’s id and the questions they created on it; group by subject.
7. Find the subject along teacher’s name, who creates questions of Subject ‘ADMS’ and display the students name and id who takes this subject. Sort them based on descending student’s id.
8. Show question id, marks and student name who answered the the questions without the student who’s name ends with ‘Rahat’. (A column should appear only once).
9. Create sequence named std\_id for new students which range (900-950), increment by 6, minvalue 550 and 10 catches.
10. Remove s\_id foreign key from result table

***Solve:***

1. SELECT s\_name, s\_phone FROM student;
2. ALTER TABLE t\_email ADD CONSTRAINT email\_unique unique(t\_email);
3. ALTER TABLE s\_phone ADD CONSTRAINT phone\_unique unique(s\_phone);
4. SELECT student.s\_name, resultTB.r\_marks FROM student, resultTB where student.s\_id = resultTB.s\_id ORDER BY student.s\_id;
5. SELECT COUNT(distinct(t\_name)) FROM teacher WHERE tname LIKE 'Rezwan%';
6. SELECT question.q\_sub, teacher.t\_id, question.q\_ques FROM teacher, question WHERE teacher.t\_id = question.t\_id ORDER BY question.q\_sub;
7. SELECT DISTINCT question.q\_sub, teacher.t\_name, student.s\_name, student.s\_id FROM question, teacher, student WHERE question.t\_id = teacher.t\_id AND question.q\_sub= 'ADMS' ORDER BY s\_id DESC;
8. SELECT DISTINCT question.q\_id, resultTB.r\_marks, student.s\_name FROM question, resultTB, student WHERE question.q\_id = resultTB.q\_id AND student.s\_id = resultTB.s\_id AND student.s\_name not like '\_%Rahat';
9. CREATE SEQUENCE student\_new\_sq START WITH 900 INCREAMENT BY 6 MAXVALUE 950 CYCLE MINVALUE 550 CACHE 10
10. ALTER TABLE resultDB DROP CONSTRAINT result\_student\_fk;

**UI Design (Login and Registration Functionality Implementation)**

***Login UI:***

***Registration UI:***

***Connection Code:***

*<?php*

*$dbname = "ExamSystem";*

*$dbuser = "system";*

*$dbpass = "orcl";*

*$connection = null;*

*try {*

*$connection = new PDO("oci:dbname=$dbname", $dbuser, $dbpass);*

*// if($connection) {*

*// echo "Connected";*

*// } else {*

*// echo "Not Connected";*

*// }*

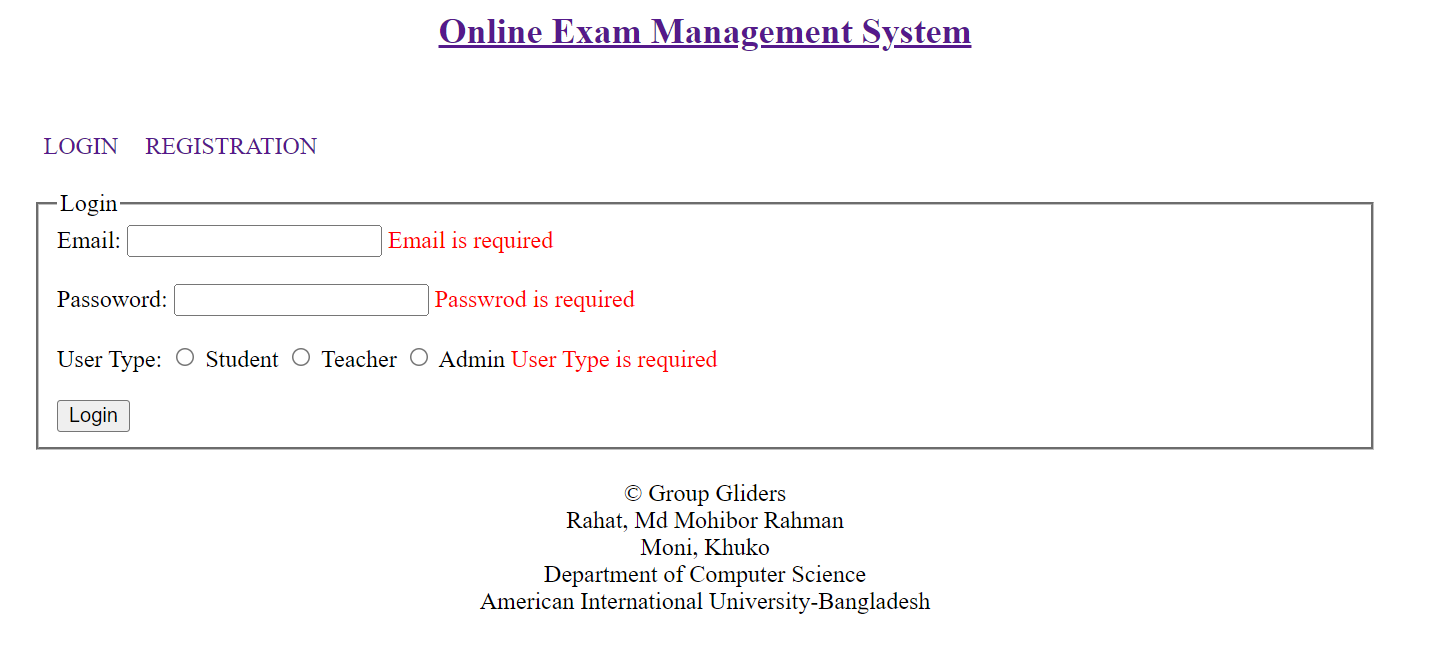
*} catch (\Throwable $th) {*

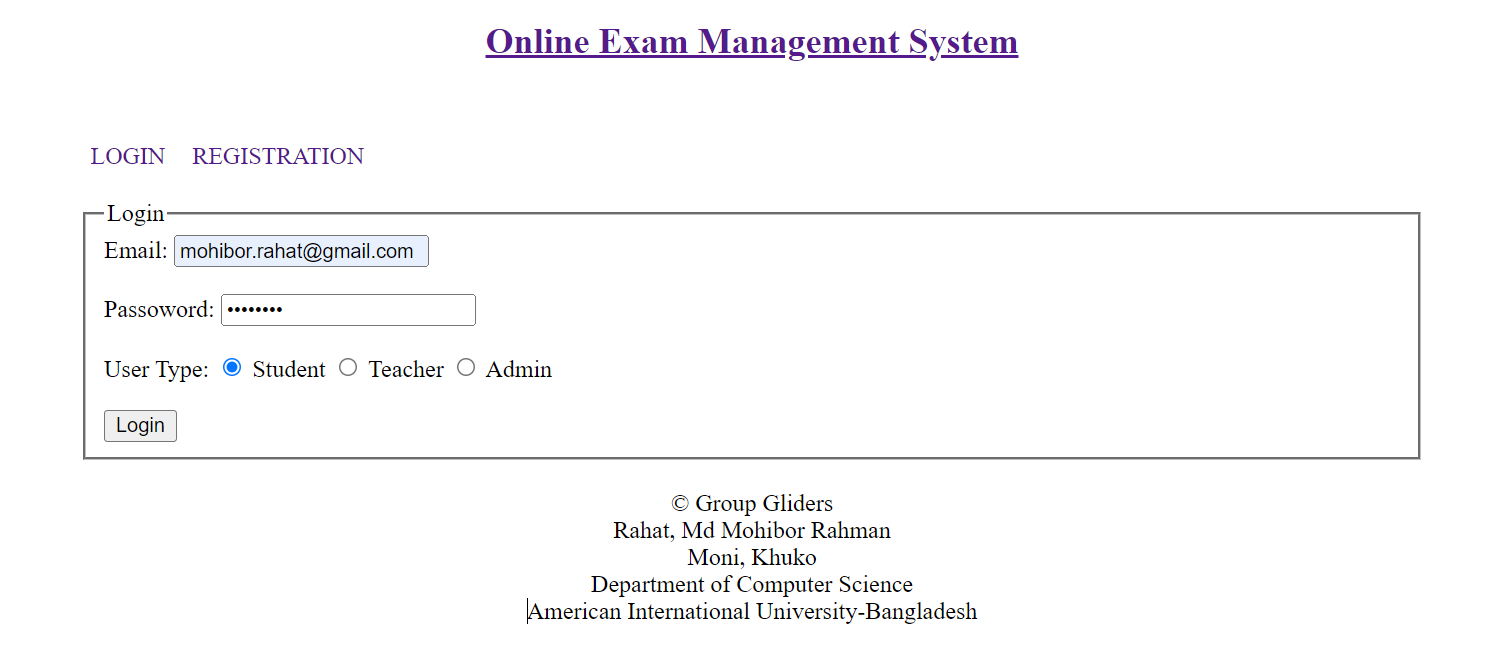
*//throw $th;*

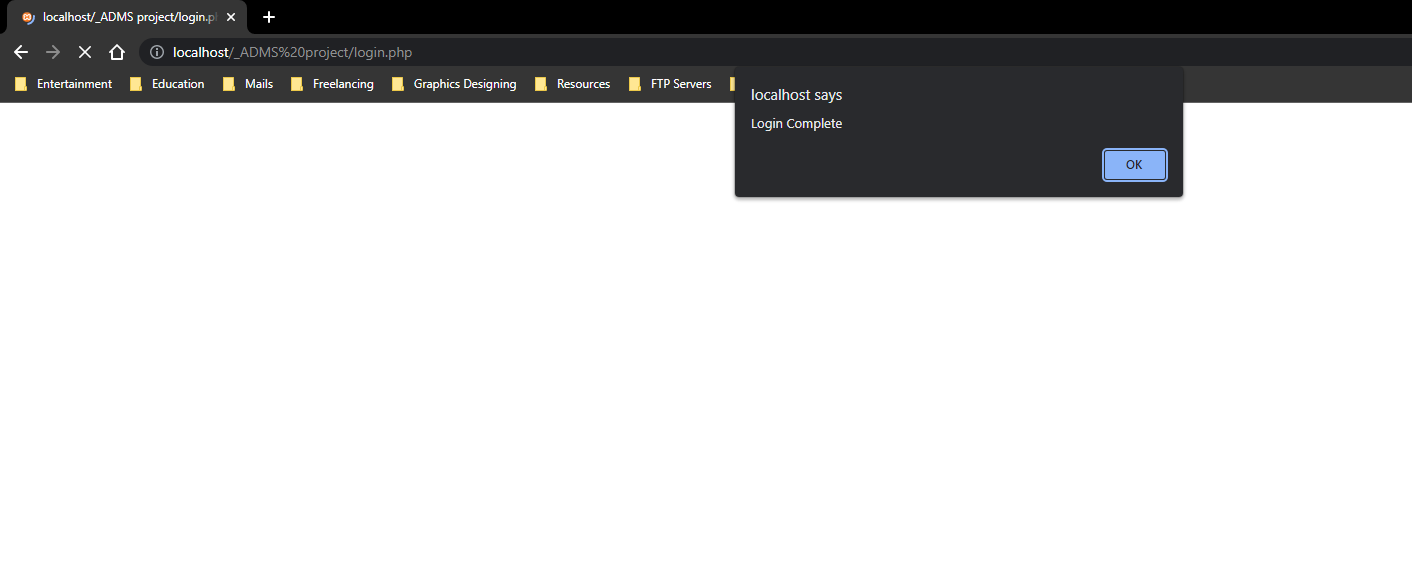
*}*

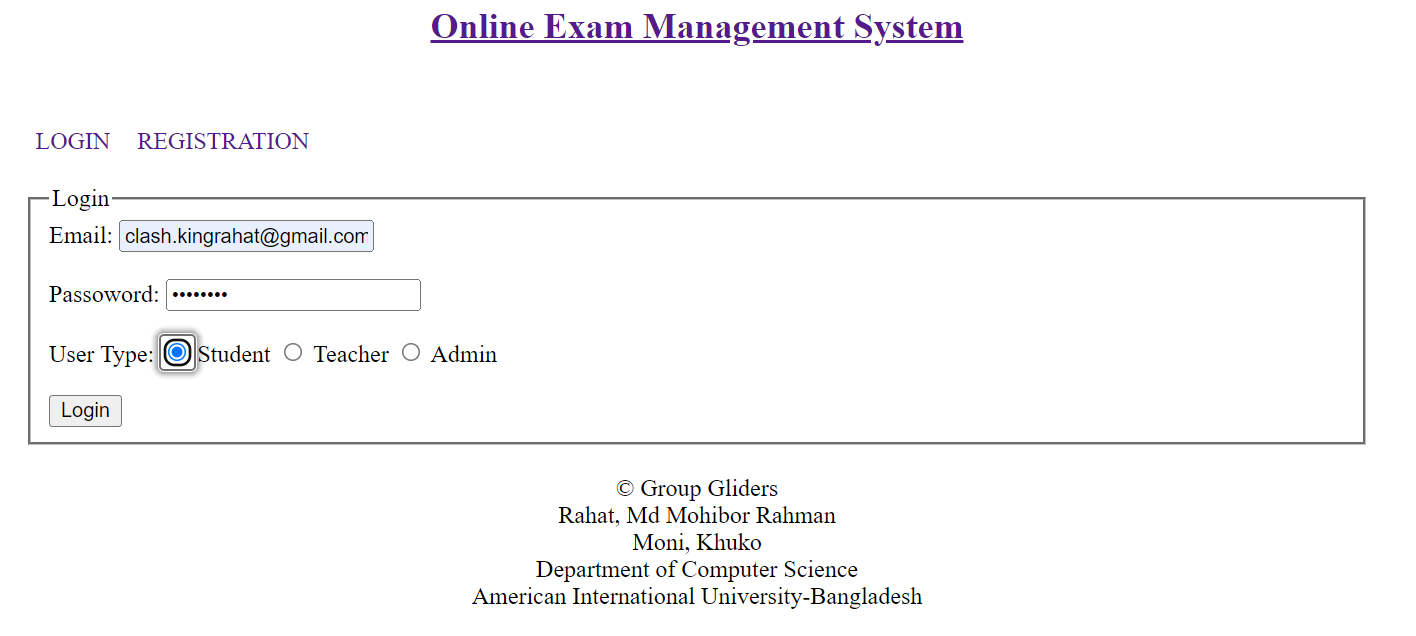
***Login Functionality:***

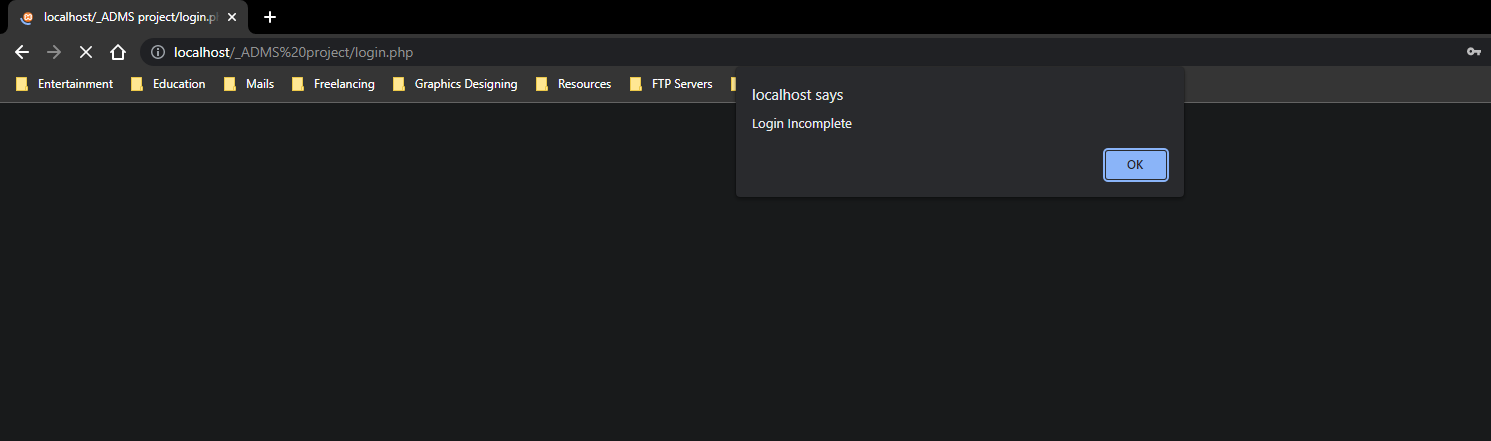
Empty Fields:



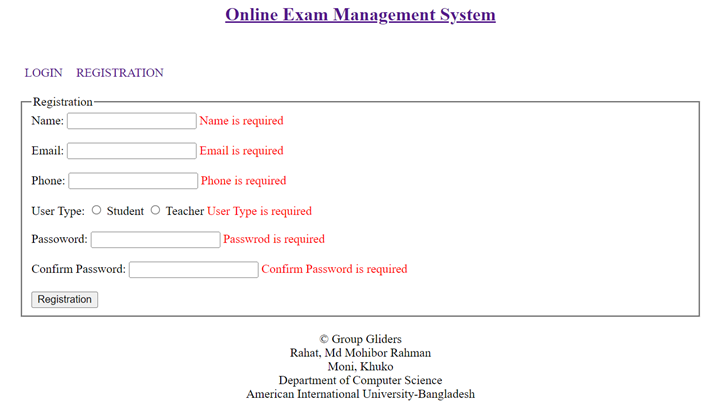
Successful Login:

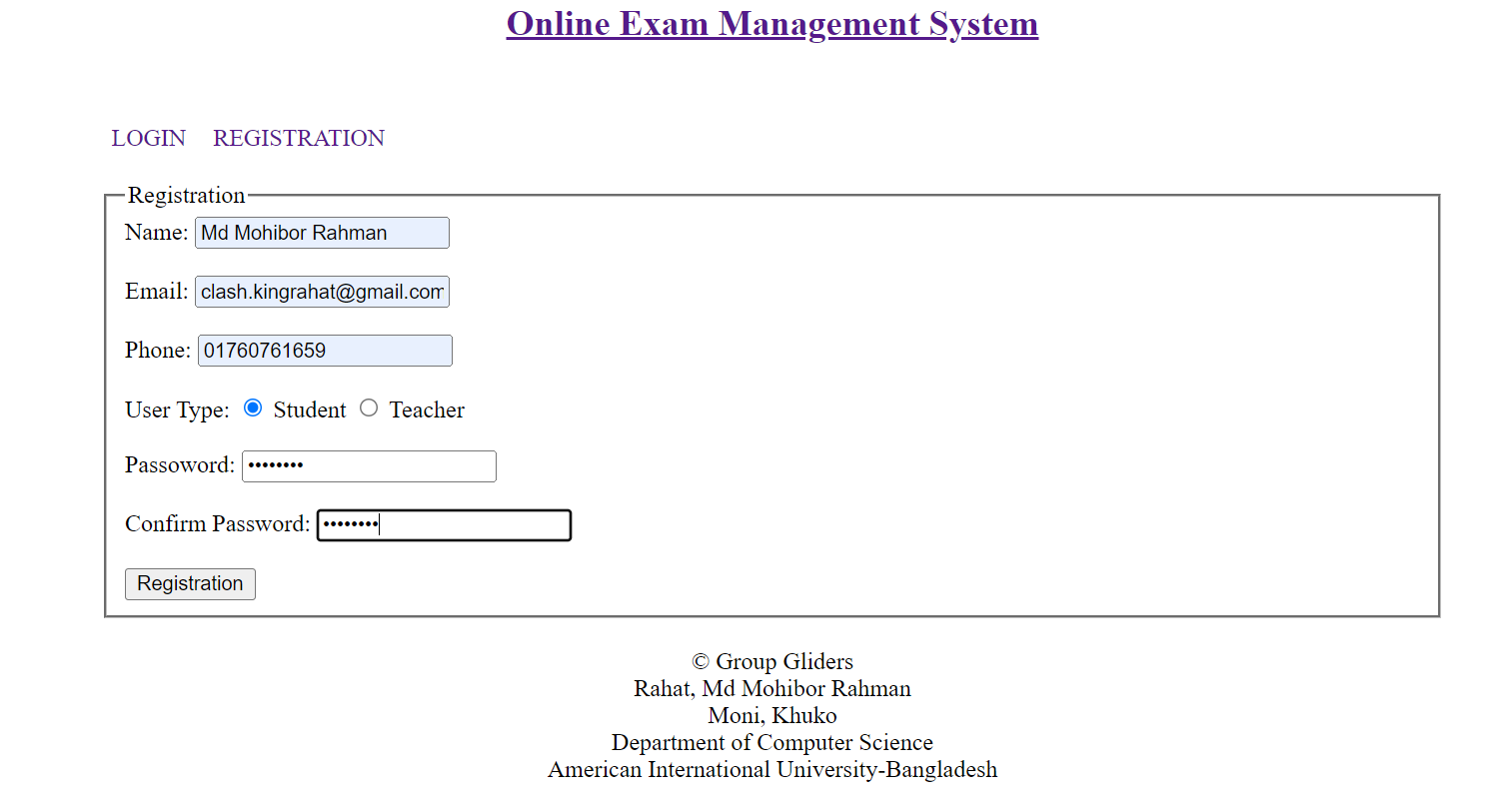


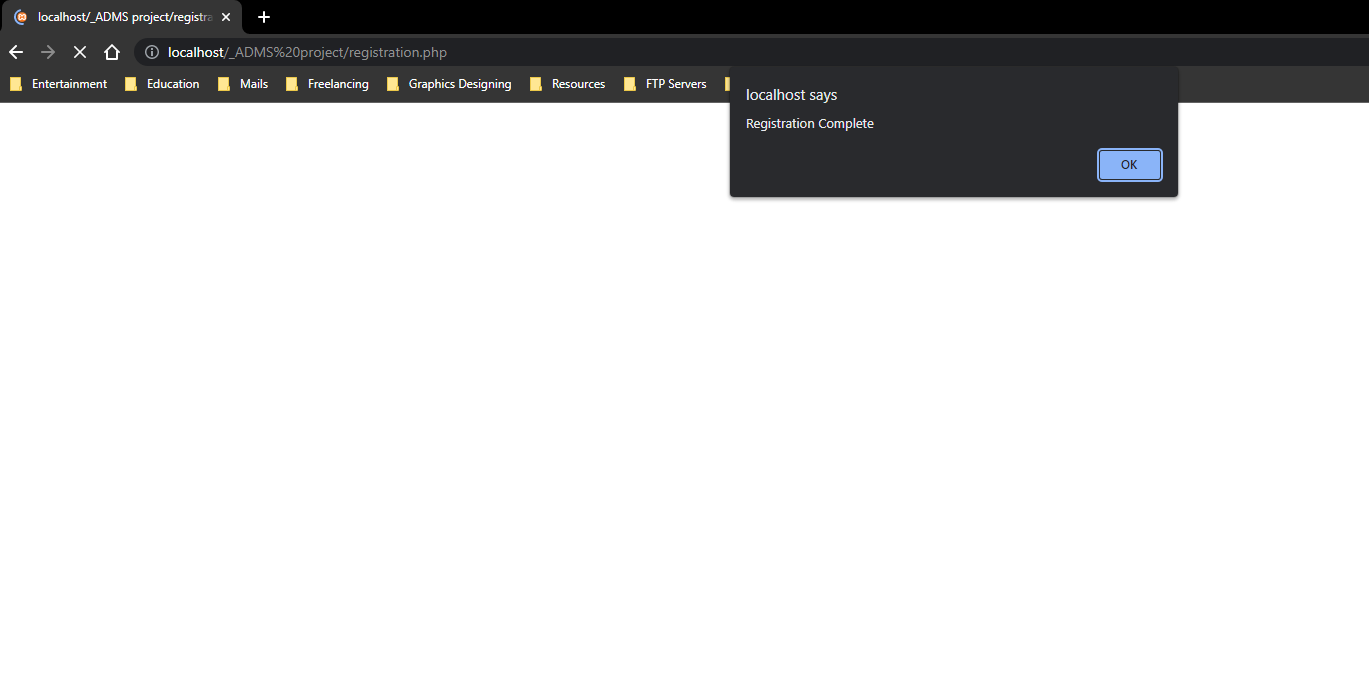
Unsuccessful Login:



***Registration Functionality:***

Empty Fields:

Successful Registration:



**Database (SQL)**

CREATE TABLE student(

s\_id NUMBER(10),

s\_name VARCHAR2(50) NOT NULL,

s\_email VARCHAR2(50) NOT NULL,

s\_phone VARCHAR2(11) NOT NULL,

s\_password VARCHAR2(50) NOT NULL

);

ALTER TABLE student ADD CONSTRAINT student\_pk PRIMARY KEY(s\_id);

CREATE SEQUENCE student\_sq START WITH 1001 INCREMENT BY 1 MAXVALUE 9999 NOCYCLE NOCACHE;

INSERT INTO student VALUES(student\_sq.NEXTVAL, 'Md. Mohibor Rahman Rahat', 'mohibor@gmail.com', '01760761659', 'password');

INSERT INTO student VALUES(student\_sq.NEXTVAL, 'Khuko Moni', 'khuko0@gmail.com', '01360761659', 'password');

INSERT INTO student VALUES(student\_sq.NEXTVAL, 'Md. Tanvir Alam Niloy', 'tniloy0@gmail.com', '01533995600', 'password');

INSERT INTO student VALUES(student\_sq.NEXTVAL, 'Rubayed Noor Shahriar', 'rubayed@gmail.com', '01999944600', 'password');

INSERT INTO student VALUES(student\_sq.NEXTVAL, 'Munem Al Shahriar', 'munem26@gmail.com', '01533107746', 'password');

CREATE TABLE teacher(

t\_id NUMBER(10),

t\_name VARCHAR2(50) NOT NULL,

t\_email VARCHAR2(50) NOT NULL,

t\_phone VARCHAR2(11) NOT NULL,

t\_password VARCHAR2(50) NOT NULL

);

ALTER TABLE teacher ADD CONSTRAINT teacher\_pk PRIMARY KEY(t\_id);

CREATE SEQUENCE teacher\_sq START WITH 1001 INCREMENT BY 1 MAXVALUE 9999 NOCYCLE NOCACHE;

INSERT INTO teacher VALUES(teacher\_sq.NEXTVAL, 'Rezwan Ahmed', 'a.rezwan@aiub.edu', '01312479154', 'password');

INSERT INTO teacher VALUES(teacher\_sq.NEXTVAL, 'Razib Hayat Khan', 'razib.hayat@aiub.edu', '01845613549', 'password');

INSERT INTO teacher VALUES(teacher\_sq.NEXTVAL, 'Rifat Tasnim Anannya', 'rifat.tasnim@aiub.edu', '01456128795', 'password');

INSERT INTO teacher VALUES(teacher\_sq.NEXTVAL, 'Abir Ahmed', 'abir.ahmed@aiub.edu', '01521432942', 'password');

INSERT INTO teacher VALUES(teacher\_sq.NEXTVAL, 'Dr. Md. Mehedi Hasan', 'mmhasan@aiub.edu', '01933340635', 'password');

CREATE TABLE adminTB(

ad\_id NUMBER(10),

ad\_email VARCHAR2(50) NOT NULL,

ad\_password VARCHAR2(50) NOT NULL

);

ALTER TABLE adminTB ADD CONSTRAINT admin\_pk PRIMARY KEY(ad\_id);

CREATE SEQUENCE admin\_sq START WITH 1001 INCREMENT BY 1 MAXVALUE 1100 NOCYCLE NOCACHE;

INSERT INTO adminTB VALUES(admin\_sq.NEXTVAL, 'admin@system.in', 'password');

INSERT INTO adminTB VALUES(admin\_sq.NEXTVAL, 'mohibor@admin.in', 'password');

INSERT INTO adminTB VALUES(admin\_sq.NEXTVAL, 'khuko@admin.in', 'password');

CREATE TABLE question(

q\_id NUMBER(10),

q\_sub VARCHAR2(255) NOT NULL,

q\_ques VARCHAR2(255) NOT NULL,

q\_ans VARCHAR2(255) NOT NULL,

q\_marks decimal(5,2) NOT NULL,

t\_id NUMBER(10) NOT NULL

);

ALTER TABLE question ADD CONSTRAINT question\_pk PRIMARY KEY(q\_id);

ALTER TABLE question ADD CONSTRAINT question\_fk FOREIGN KEY (t\_id) REFERENCES teacher(t\_id);

CREATE SEQUENCE question\_sq START WITH 1001 INCREMENT BY 1 MAXVALUE 9999 NOCYCLE NOCACHE;

INSERT INTO question VALUES(question\_sq.NEXTVAL, 'ADMS', 'What''s the full form of ADMS?', 'Advance Database Management System', 10, 1001);

INSERT INTO question VALUES(question\_sq.NEXTVAL, 'ADMS', 'What''s the full form of RDBMS?', 'Relational Database Management System', 10, 1001);

INSERT INTO question VALUES(question\_sq.NEXTVAL, 'ADMS', 'What are the types of normalization in database?', '1NF, 2NF, 3NF', 10, 1001);

INSERT INTO question VALUES(question\_sq.NEXTVAL, 'ADMS', 'What''s the full form of DDL?', 'Data Definition Language', 10, 1001);

INSERT INTO question VALUES(question\_sq.NEXTVAL, 'ADMS', 'What''s the full form of DML?', 'Data Manipulation Language', 10, 1001);

CREATE TABLE resultTB(

r\_id NUMBER(10),

r\_marks DECIMAL(5, 2) DEFAULT 0.00 NOT NULL,

s\_id NUMBER(10) NOT NULL,

q\_id NUMBER(10) NOT NULL

);

ALTER TABLE resultTB ADD CONSTRAINT result\_pk PRIMARY KEY(r\_id);

ALTER TABLE resultTB ADD CONSTRAINT result\_student\_fk FOREIGN KEY (s\_id) REFERENCES student(s\_id);

ALTER TABLE resultTB ADD CONSTRAINT result\_ques\_fk FOREIGN KEY (q\_id) REFERENCES question(q\_id);

CREATE SEQUENCE result\_sq START WITH 1001 INCREMENT BY 1 MAXVALUE 9999 NOCYCLE NOCACHE;

INSERT INTO resultTB VALUES(result\_sq.NEXTVAL, 10, 1001, 1001);

INSERT INTO resultTB VALUES(result\_sq.NEXTVAL, 10, 1002, 1002);

INSERT INTO resultTB VALUES(result\_sq.NEXTVAL, 10, 1003, 1003);

INSERT INTO resultTB VALUES(result\_sq.NEXTVAL, 10, 1004, 1004);

INSERT INTO resultTB VALUES(result\_sq.NEXTVAL, 10, 1005, 1005);

CREATE TABLE approval(

ap\_id NUMBER(10),

ap\_name VARCHAR2(50) NOT NULL,

ap\_email VARCHAR2(50) NOT NULL,

ap\_phone VARCHAR2(11) NOT NULL,

ap\_password VARCHAR2(50) NOT NULL,

ap\_usertype VARCHAR2(50) NOT NULL

);

ALTER TABLE approval ADD CONSTRAINT approval\_pk PRIMARY KEY(ap\_id);

CREATE SEQUENCE approval\_sq START WITH 1001 INCREMENT BY 1 MAXVALUE 9999 NOCACHE;

INSERT INTO approval VALUES(approval\_sq.NEXTVAL, 'Md Mosabbir Jayed', 'mosabbir.jayed@gmail.com', '01944600486', 'password', 'student');

INSERT INTO approval VALUES(approval\_sq.NEXTVAL, 'Sifat Rahman Ahona', 'ahona@aiub.edu', '01956781954', 'password', 'teacher');

INSERT INTO approval VALUES(approval\_sq.NEXTVAL, 'Fahmida Alam', 'fahmida@aiub.edu', '01845278319', 'password', 'teacher');

INSERT INTO approval VALUES(approval\_sq.NEXTVAL, 'Nahian Sajjad', 'nahian@gmail.com', '01578126489', 'password', 'student');

INSERT INTO approval VALUES(approval\_sq.NEXTVAL, 'Rumman Rafty', 'rumman@gmail.com', '01360761659', 'password', 'student');

commit;