

COMP1640: Enterprise Web Software Development

EC_Claims Coursework Report

Submitted Date

13th April 2017

Submitted To

COORDINATOR: Dr Ray Stoneham

University of Greenwich

Submitted By

by Md. Johirul Islam, ID: 000989744

Title Page

Team Name: Group-1

Team Members and Roles:

Name	Role
Md. Johirul Islam	Database Development
Abhijit Barua	System Analyst/Scrum Master
Md. Tariqul Islam	Programmer/UI Designer
Mukta Akter	Tester



Username and Password

Role	Username	Password
System Admin		
EC Coordinator		
EC Manager		
Student		

Table of Content

Title Page	2
Table of content.....	3
Evaluation	4
Meeting the system requirements.....	4
Further Development	13
Self-Contribution	13
My Contribution	13
Lesson Learnt	14
Team Contribution	14
Used Methodology	15
Conclusion	15
Appendices:	
 Appendix A : Database Design	16
Normalisation Process	16
Extended Entity Relationship Diagram	18
Data Mapping	19
Data Dictionary	20
Implementation of Integrity Requirements	24
Retrieving Data	30
Security Measures	32

Evaluation

Meeting the System's Requirements:

After complete a wide range of analysis, we have developed our EC Claim Submission System. We properly fulfill all requirements of our client. For developing the system properly our team members gave their full efforts. For this, we arranges meeting regularly. A briefly description of the system requirements and how our system has delivered those requirements are stated below:

- The coursework required a role based web application and there will be different types of user. Our system is developed using PHP, MySQL and JavaScript. In addition, our user interface designed using HTML, CSS and Bootstrap. In our system, there have different types of limited dashboard for different user. They can only complete their own task.
- The coursework required different types of user like Admin, EC Manager, Course Coordinator, Student and guest. Our system also has all types of role.

From the following picture you can understand the below pictures:

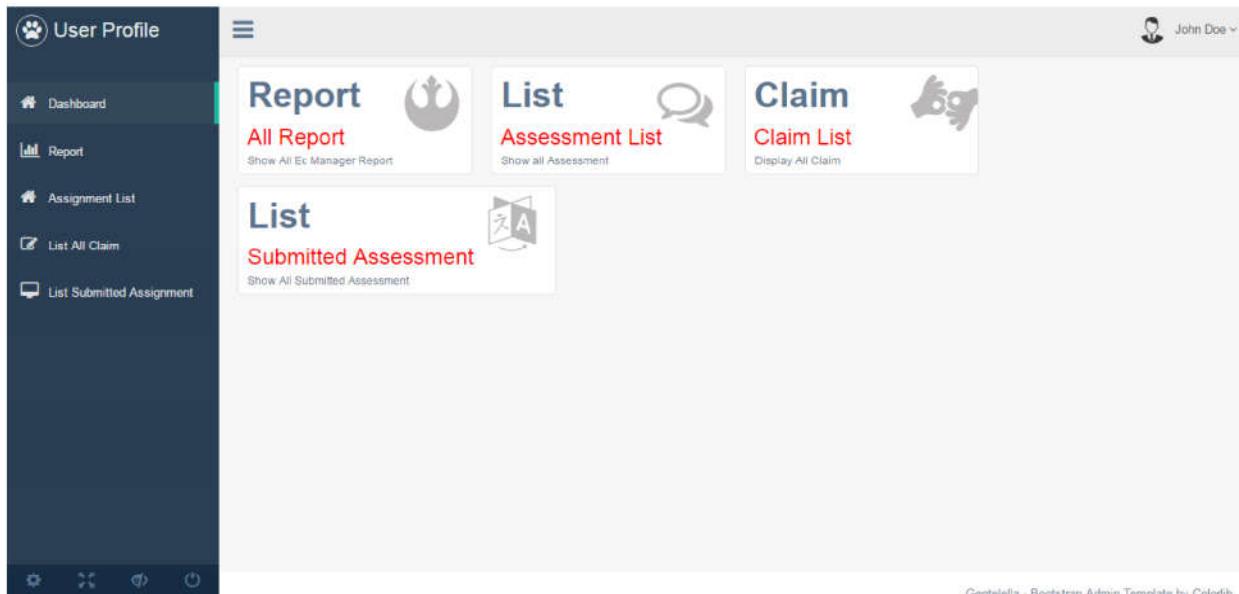


Figure: Dashboard for the EC Manager role

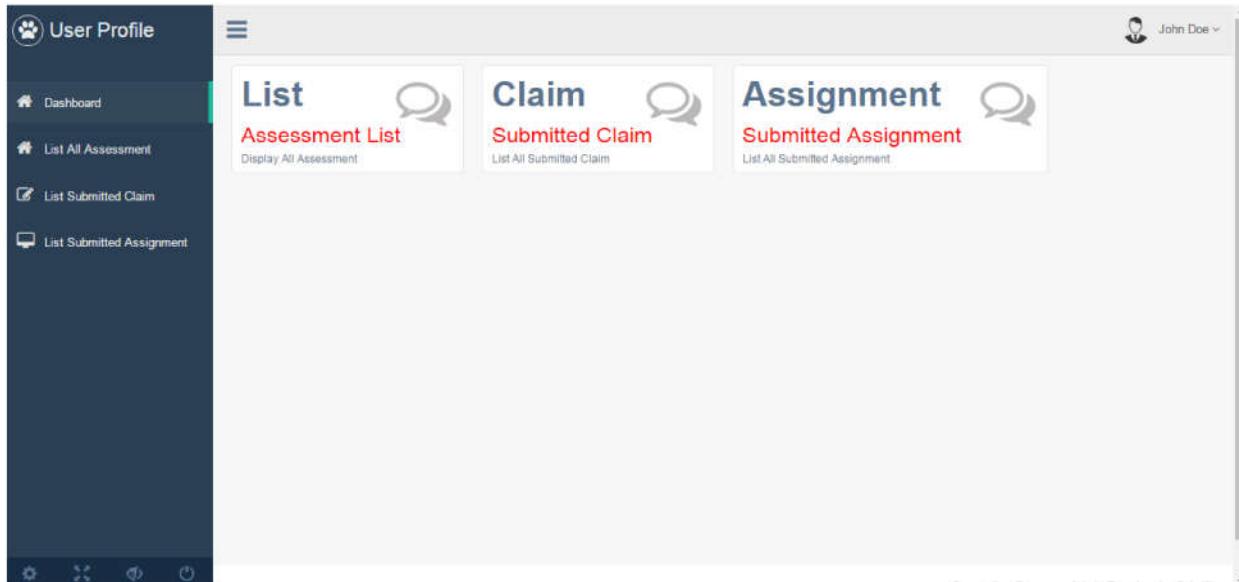


Figure: Dashboard for Faculty Coordination

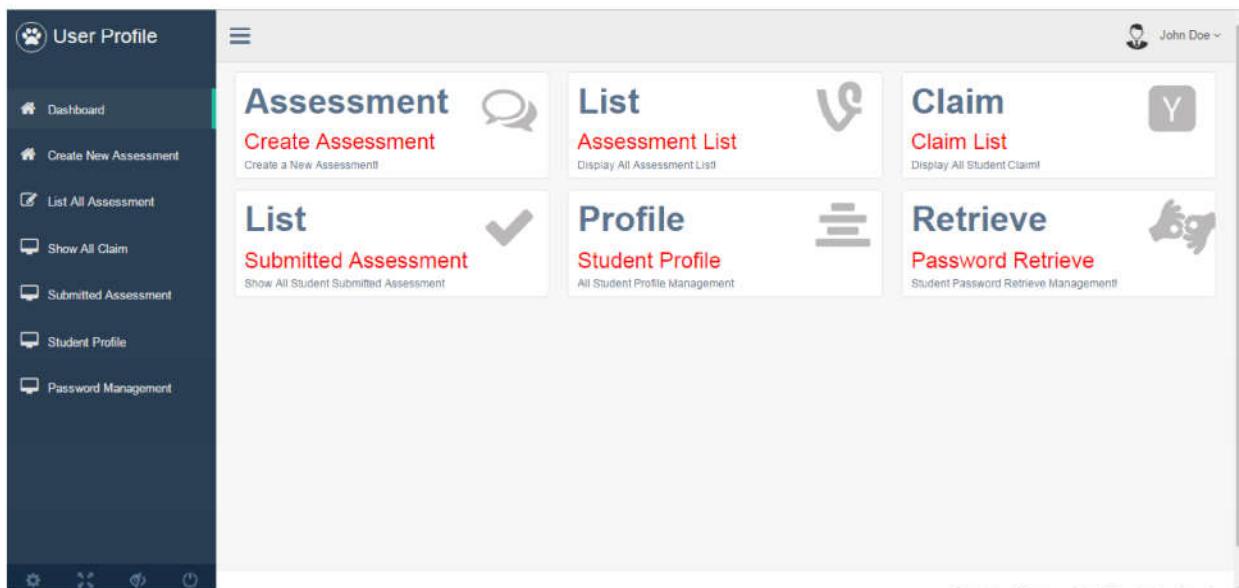


Figure: Dashboard for Administration

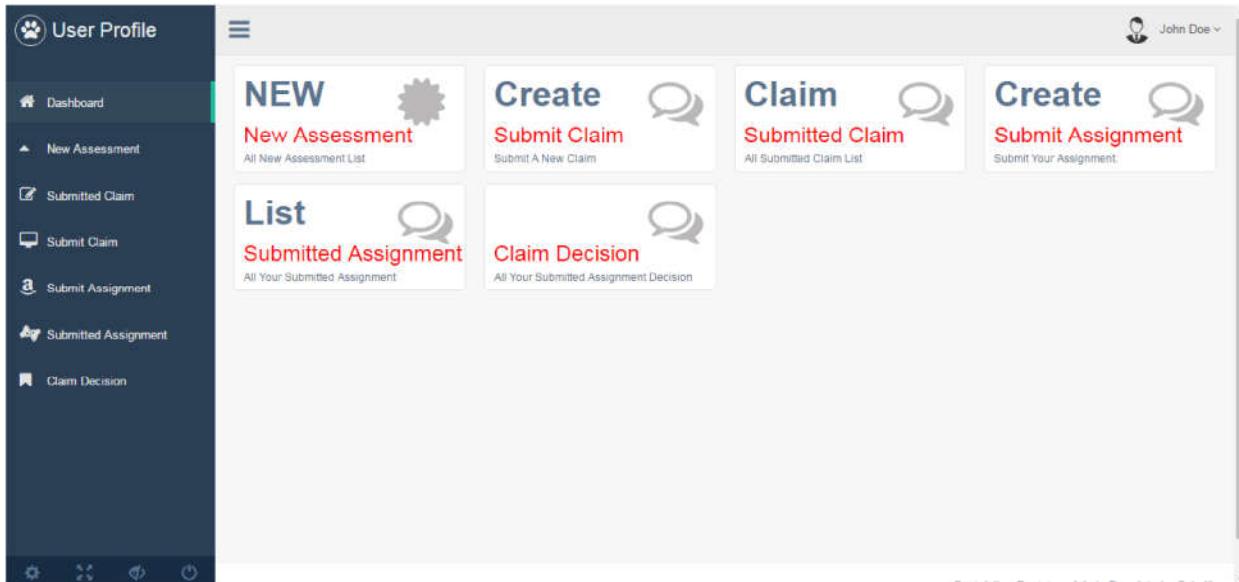


Figure: Dashboard for students

After login by their email and password, system will automatically find out the of this email and sent him into his dashboard based on email. From there, they can only complete their own task. From the above dashboard, we can see the differences between dashboards

- Our system is fully responsive and compatible for both computer and mobile screen.

The screenshot shows a 'Create A New Assessment' form. The sidebar on the left includes 'Create New Assessment' (selected), 'List All Assessment', 'Show All Claim', 'Submitted Assessment', 'Student Profile', and 'Password Management'. The main form has fields for Faculty Name (Science), Department Name (IT), Assessment Name, Batch Number, Academic Session, Submission Last Date, and Late Submission Date. There is also an 'Upload Assignment Doc' button and three buttons at the bottom: 'Cancel', 'Reset', and 'Submit'.

Figure: Site responsiveness during full screen window

Department Name *

Assessment Name *

Batch Number *

Academic Session *

Submission Last Date *

Late Submission Date: *

Upload Assignment Doc

[Cancel](#) [Reset](#) [Submit](#)

Gentelella - Bootstrap Admin Template by Colorlib

Figure: Site responsiveness window size is half

Therefore, from those pictures we can say that our site is completely responsive.

Our site is also responsible in mobile device. It will remain responsive if it use in any other device.

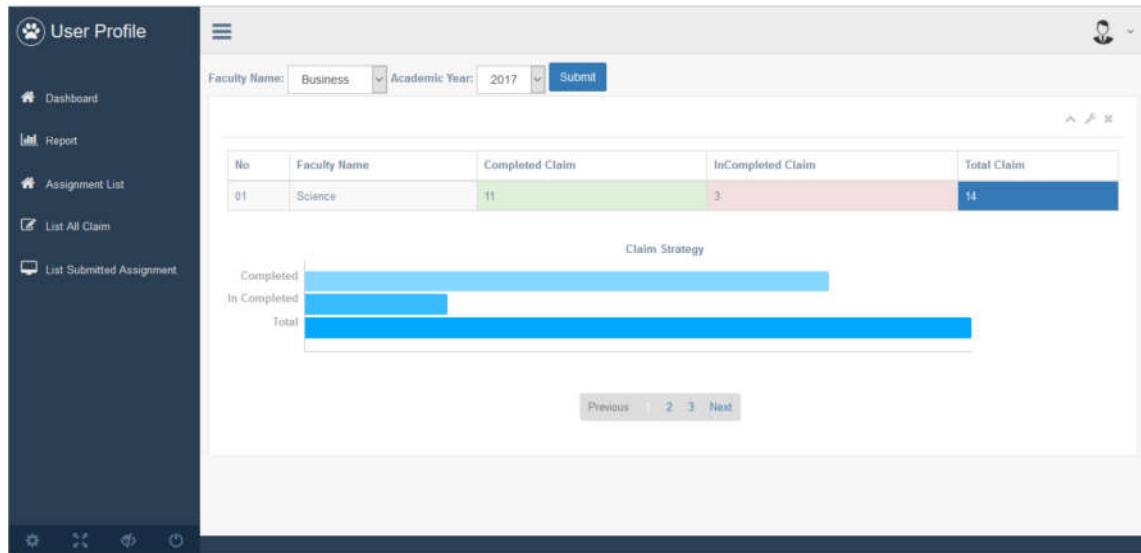


Figure: Report page in Computer browser

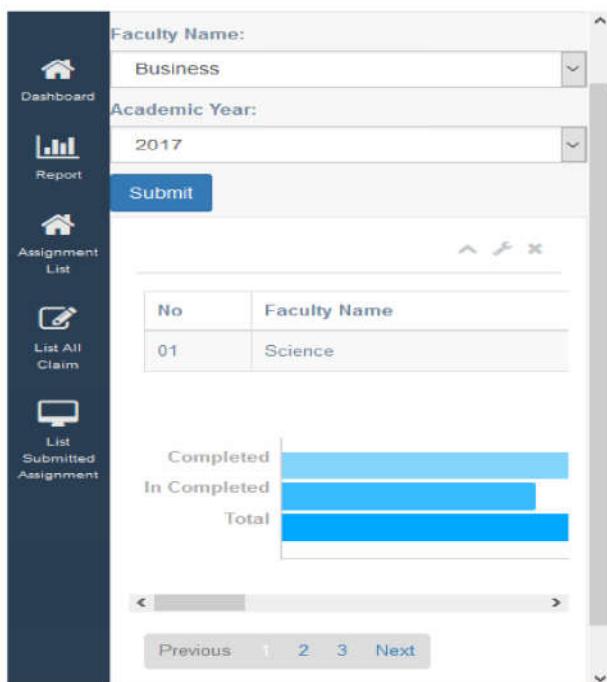


Figure: Report page in Mobile browser

The screenshot displays a mobile responsive web application interface. On the left, a dark sidebar contains icons and labels for 'Dashboard', 'Create New Assessment' (which is highlighted in green), 'List All Assessment', 'Show All Claim', 'Submitted Assessment', and 'Student Profile'. The main content area has a header 'Create A New Assessment'. It features several input fields with dropdown menus: 'Faculty Name *' (set to 'Business'), 'Department Name *' (set to 'BBA'), 'Assessment Name *' (empty), 'Batch Number *' (empty), and 'Academic Session *' (empty). At the bottom of the form, there is a small note: 'Gentelella - Bootstrap Admin Template by' followed by a link.

Figure: Create new assessment form in mobile browser

After seeing the above screenshots, we can said that our site is proper responsive in any browser and in both PC and Mobile device.

- In our system, Admin can control the whole system and the EC manager can only monitor all information. Admin can upload assessment file, maintain user account, password and more that mentioned in the coursework.

- In the system, user can submit claim with or without evidence. If they submit claim without evidence then it will be automatically elimination after 14 days and they need to submit their claim within assignment submitted date. All of these were also in requirements list.

Create your claim

Assessment Name	EWS
Claim Title *	I was very ill
Claim Description	It was not my mistake It was not my mistake It was not my mistake

Upload Your Evidence

Cancel **Reset** **Submit**

Figure: Submitting Claim without evidence

Edit your claim

Assessment Name	New Assessment
Claim Title *	I was very ill
Claim Description	It was not my mistake It was not my mistake It was not my mistake

Change add.jpg ×

Cancel **Reset** **Submit**

Figure: Submit claim with evidence

All Submitted Claim List						
Show <input type="text" value="10"/> entries		Search: <input type="text"/>				
No	Assessment Name	Title	Modify Date	Status	Operation	
11	New Assessment	Hello USA	2017-03-26	With Evidence Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
12	New Assessment	Hello USA	2017-03-26	Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
13	dg	Hello USA	2017-04-08	Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
14	dg	Hello USA	2017-04-08	Without Evidence Incomplete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
15	dg	football	2017-04-08	Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
16	New Assessment	Nice-2	2017-04-08	Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
17	DW	Claim For DE	2017-04-09	Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
18	New Assessment	Hello	2017-04-09	Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
19	EWSD	Leg broken problem	2017-04-11	Complete	<input type="button" value="View"/>	<input type="button" value="Edit"/>
20	EWSD	I was very ill	2017-04-11	Incomplete	<input type="button" value="View"/>	<input type="button" value="Edit"/>

Figure: Claim list with status

If student submit claim without evidence then it will show incomplete or if student provide all required information included evidence then it show complete.

- The requirement also stated that after submitting claim notified the EC Coordinator and after accepting or rejecting claim notified the student via email notification. The system also meets both requirements.

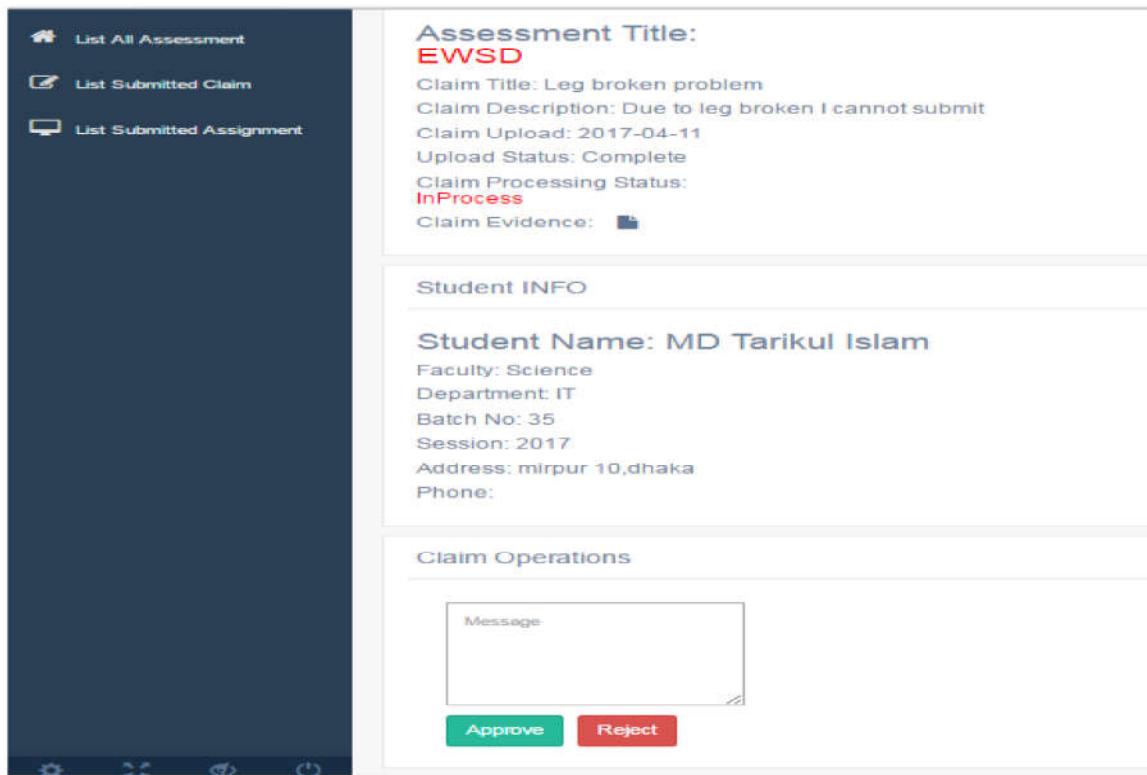


Figure: Approve or reject claim by Course coordinator

According to the agile methodology, at first we deliver the important features to make the system functional. We use Scrum to make our work easier. Therefore, we can say that our system is fully secure and meets all the requirements that stated in the coursework.

Further Development:

Although our system meets all the requirements, our system still has some weaknesses which need to solve and some features need to add in future. In our system EC Manager, can only view the system, in future, we give him an option for give his opinion and notify him automatically whenever anything happens. In future admin can also add any necessary function automate. In our system, claims and assignment are still visible after due date. Our dashboards will be more informative and beautiful. For that, we will add more statistical report that generate and provide information automatically. We also improve our assignment and claim submission system. There will add message system during assignment so that student can chat with their teacher and student course material will provide through our website. We will also make our system more flexible and beautiful for all users. We also survey for user experience to make our system more usable.

Self-Contribution:

My contribution:

In this team, my role is as a Database Designer. However, I also contribute with UI design. My responsibilities start before the development process start. I directly worked with the programmer so that he could understand the database and its relationship properly.

After starting the responsibility, I analysis the whole coursework and normalized the whole scenario and assume entities and attributes. For doing my work properly, I took help from our analyst. I got functional and non-functional requirements from our analyst. He completely described me what type of database I needed to make. In our project, we use agile scrum methodology. I discuss with our team members during building the database. We make short meeting regularly and a weekly long meeting.

Briefly explanation of my working process:

- After getting all requirements from the analyst and deeply understanding the requirements, I normalized all data. For this at first, I make 1NF to identify all attribute and remove repeating group information. Then I make 2NF to remove partial key dependencies. Finally, I make 3NF to remove non-key dependencies and create foreign keys to create relationship properly.
- After that, I make an Extended Entity Relationship Diagram for the system.
- I also create a Data Mapping so that I create database easily and understand all the relationship.
- Then I make data dictionary for my system where I define all attributes data type, foreign key and other necessary things. I also create Indexes for all the tables.

- By discussing our team, we also decided to develop our database using MySQL. This because our project is building via PHP and MySQL is good for PHP.
- Then I developed the database. I also help the programmer by making some queries during the system development time.

During the system development time, I was also involved with our team members. We complete our system by contributing with each other. I was joining in every team meeting. In addition, give my opinion for developing the system properly.

Lesson Learnt:

Although this is not my first teamwork, I have learnt many new things from this system development. This is first time I have used Scrum methodology and I have a proper idea about the Scrum methodology. I also understand the benefits of this methodology. From this project, I got a clear understanding about the role based system development. I learnt how to take pressure to complete my own task within a given time. Therefore, for me, it was a great learning experience.

Team contribution:

The name of our team is Group-1 where we are four team members. I think our team was a great balanced team in where every try their best to complete their responsibilities and make the system more functional. Each member was in charge of different section so that we developed our system quickly. After deeply analysis the requirements, we make a clear concept about the requirements. We regularly communicate with each other via mobile, Skype or Facebook and discuss about the system. After analyzed the system properly, I developed the database properly and the programmer and UI designer showed outstanding performance to make system functional and user friendly. For reaching the final output, we faced multiple iterations of the system. Our team members gave their full effort to reach the final output that we desired. After developing every part of the system, our tester tested the system and tried to find out mistakes or better improvements. Hence, every members of our team gave their full effort to develop the system properly.

Assessing Team Performance

Team Member Role	Assessment	Performance rating (out of 10)
System Analyst	Excellent Performance	9
Database Designer	Excellent Performance	9
Programmer & UI designer	Excellent Performance	9
System Tester	Excellent Performance	8

Used Methodology:

In our group work, we wanted to adopt agile methodology. However, there are several types of agile Framework. From all of them, we choose the best agile framework called Scrum. This is because is more flexible and smarter way of developing a system. Scrum also supports group work and makes the work more easy and quick. We made our system in five sprints that help us to deliver the system in time. Iterative and incremental development made the framework more popular. In our group, each member was assign for completing their own task in an ordered time. During each sprint, after completing our own task, we discuss with each other in our group meeting, finding mistakes and discuss for further development.

Conclusion:

The coursework required from our team to complete a “Student Claim Management System” and we deliver it within the given time that was not too easy. We faced many difficulties during the development time. After given full 2 months effort we had completed our system. Despite of some lack, we had completed our system in time and we hope that we will overcome those difficulties and mistakes in future.

Appendix: A (Database Design)

1.1 Normalisation Process:

1NF => 1st Normalisation

2NF => 2nd Normalisation

3NF => 3rd Normalisation

1NF	2NF	3NF
<u>Users</u> id email password departmentName facultyName session batch_no role st_name st_dob st_email st_phone st_address st_photo	<u>Academic</u> id email password departmentName facultyName session batch_no <u>Management student</u> id email password role st_name st_dob st_email st_phone st_address st_photo	<u>Department</u> id department_name <u>Faculty</u> id facultyName department_name <u>Academic session</u> id batch_no session department_name <u>Management</u> id email password role faculty <u>Student</u> id st_name st_dob st_email st_phone st_email st_phone st_address st_photo st_password department batch_no
<u>Assesment claim</u> ass_name ass_dates duDate lastSubDate ass_document	<u>Assessment assignment</u> id ass_name ass_dates duDate lastSubDate	<u>Assessment</u> id ass_name sessions batch_no dates

upload_date upload_document ass_description ec_title ec_description ec_upload_date processing_status ec_documents claim_status pass_confirmation notification_title senderID receiverID	ass_document upload_date upload_document ass_description	duDate lastSubDate documents facultyName
		Assignment assignment_id title description upload_date upload_document department_name ass_id
	Claim_password_info ec_title ec_description ec_upload_date processing_status ec_documents claim_status pass_confirmation notification_title senderID receiverID	Claim ec_id title description upload_date upload_document claim_status st_id ass_id
		Notification id title description senderID receiverID
		Password_retrive id email management

1.2 Extended Entity Relationship Diagram

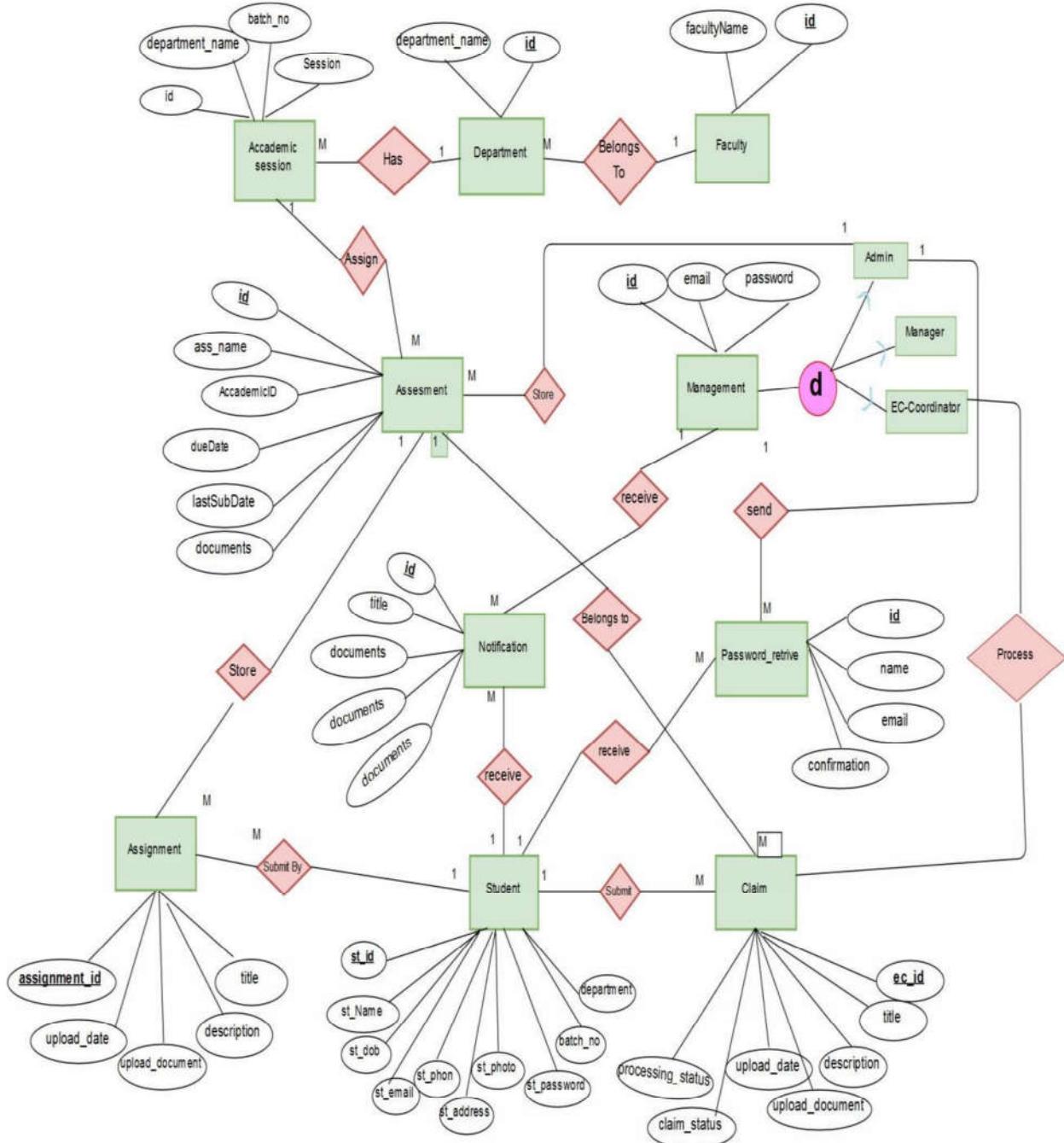


Figure: Extended Entity Relationship Diagram

1.3 Data Mapping:

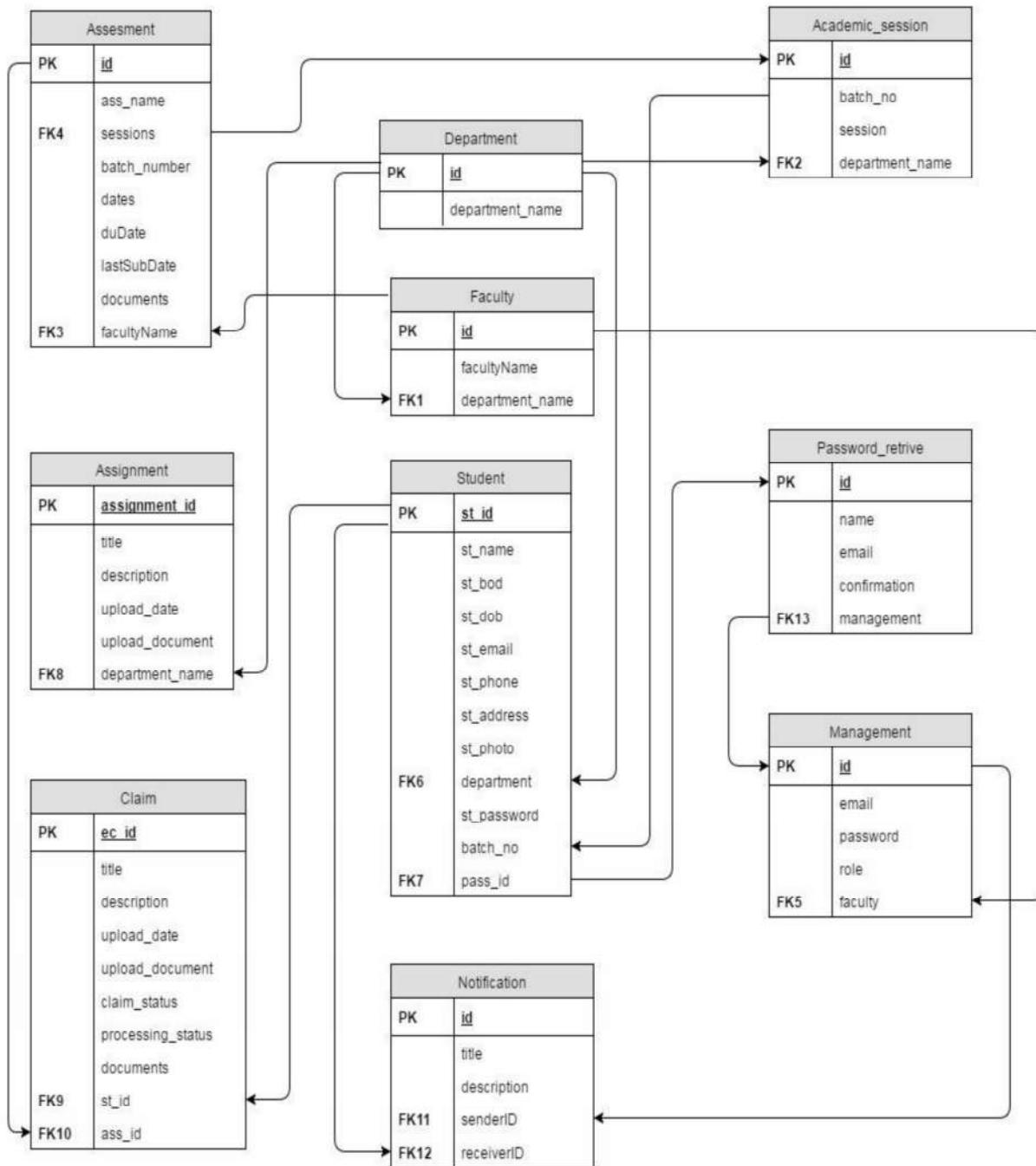


Figure: Data mapping

1.4 Data Dictionary:

➤ **Assesment**

Column	Type	Null	Default	Links to	Comments
id (Primary)	int(20)	No			
ass_name	varchar(200)	Yes	Null		
facultyName	varchar(200)	Yes	Null		
batch_number	varchar(100)	No			
sessions	varchar(100)	No			
dates	date	Yes	Null		
duDate	date	Yes	Null		
lastSubDate	date	Yes	Null		
documents	text	No			
departmentName	varchar(100)	No		department ->department_name	

Indexes:

Keyname	Type	Uniq ue	Packe d	Column	Cardinalit y	Collatio n	Nul l	Comm ent
PRIMARY	BTREE	Yes	No	id	0	A	No	
department Name	BTREE	No	No	departmen tName	0	A	No	

➤ **Assignment**

Column	Type	Null	Default	Links to	Comments
ass_id (primary)	int (11)	No			
title	varchar(100)	No			
description	varchar(100)	No			
upload_document	text	No			
upload_documents2	text	No			
upload_date	date	No			
assesment_Id	varcahr(100)	No		assessment-> id	
student_email	varcahr(100)	No		student-> st_email	

Indexes:

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	ass_id	0	A	No	
assessment_id	BTREE	No	No	assessment_id	0	A	No	
student_email	BTREE	No	No	student_email	0	A	No	

➤ Student

Column	Type	Null	Default	Links to	Comments
st_id (PRIMARY)	int(20)	No			
st_FirstName	varchar(100)	Yes	Null		
st_LastName	varchar(100)	Yes	Null		
st_dob	date	Yes	Null		
st_email	varchar(100)	Yes	Null		
st_phon	varchar(30)	Yes	Null		
st_address	varchar(200)	Yes	Null		
st_photo	varchar(200)	Yes	Null		
st_password	varchar(30)	Yes	Null		
batchNo	varchar(30)	Yes	Null		
session	varchar(100)	Yes	Null		
Faculty	varchar(100)	Yes	Null		
department	varchar(100)	No		department-> department_name	

Indexes:

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	st_id	0	A	No	
department	BTREE	No	No	department	0	A	No	

➤ Department

Column	Type	Null	Default	Links to	Comments
id (PRIMARY)	int(11)	No			
department_name	varchar(100)	No			
facultyID	int(11)	No			faculty->id

Indexes:

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	0	A	No	
facultyID	BTREE	Yes	No	facultyID	0	A	No	

➤ Faculty

Column	Type	Null	Default	Links to	Comments
id (PRIMARY)	int(11)	No			
faculatynName	varchar(100)	No			

Indexes:

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	0	A	No	

➤ Claim

Column	Type	Null	Default	Links to	Comments
ec_id	int(20)	No			
title	varchar(500)	Yes	Null		
description	varchar(1000)	Yes	Null		
upload_document	varchar(500)	Yes	Null		
upload_document2	varchar(500)	Yes	Null		
upload_document3	varchar(500)	Yes	Null		
upload_document4	varchar(500)	Yes	Null		
upload_date	date	Yes	Null		
assessment_Id	int(20)	No			assessment-> id
claim_status	varchar(500)	Yes	Null		
processing_status	varcahr(100)	No			
student_email	varchar(100)	No			student-> st_email

Indexes:

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTRE E	Yes	No	id	0	A	No	
assessment_id	BTRE E	No	No	assessment_id	0	A	No	
student_email	BTRE E	No	NO	student_email	0	A	No	

Management

Column	Type	Null	Default	Links to	Comments
id <i>(PRIMARY)</i>	int(11)	No			
email	varchar(100)	No			
password	varchar(100)	No			
role	varchar(100)	No			
facultyName	varchar(100)	No		faculty-> facultyName	

Indexes:

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTRE E	Yes	No	id	0	A	No	
facultyName	BTRE E	No	No	facultyName	0	A	No	

1.4 Implementation of Integrity Requirements

The SQL code for creating all the tables and their referential constraints and respective integrity are giving below:

Assesment:

```
CREATE TABLE IF NOT EXISTS `assesment` (
  `id` int(20) NOT NULL AUTO_INCREMENT,
  `ass_name` varchar(200) DEFAULT NULL,
  `facultyName` varchar(200) DEFAULT NULL,
  `batch_number` varchar(100) NOT NULL,
  `sessions` varchar(100) NOT NULL,
  `dates` date DEFAULT NULL,
  `duDate` date DEFAULT NULL,
  `lastSubDate` date DEFAULT NULL,
  `documents` text NOT NULL,
  `departmentName` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=28 ;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** student
- Table:** assesment
- Table Structure View:** The table has 10 columns: id, ass_name, facultyName, batch_number, sessions, dates, duDate, lastSubDate, documents, and departmentName.
- Column Details:**
 - id:** Type int(20), Collation utf8mb4_unicode_ci, Attributes AUTO_INCREMENT, Null No, Default NULL.
 - ass_name:** Type varchar(200), Collation utf8mb4_unicode_ci, Attributes Null Yes, Default NULL.
 - facultyName:** Type varchar(200), Collation utf8mb4_unicode_ci, Attributes Null Yes, Default NULL.
 - batch_number:** Type varchar(100), Collation utf8mb4_unicode_ci, Attributes Null No, Default None.
 - sessions:** Type varchar(100), Collation utf8mb4_unicode_ci, Attributes Null No, Default None.
 - dates:** Type date, Collation utf8mb4_unicode_ci, Attributes Null Yes, Default NULL.
 - duDate:** Type date, Collation utf8mb4_unicode_ci, Attributes Null Yes, Default NULL.
 - lastSubDate:** Type date, Collation utf8mb4_unicode_ci, Attributes Null Yes, Default NULL.
 - documents:** Type text, Collation utf8mb4_unicode_ci, Attributes Null No, Default None.
 - departmentName:** Type varchar(100), Collation utf8mb4_unicode_ci, Attributes Null No, Default None.
- Action Bar:** Includes buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers.
- Bottom Buttons:** Print view, Propose table structure, Track table, Move columns, Improve table structure.

Figure: Screenshot for Assessment table

Assignment:

```
CREATE TABLE IF NOT EXISTS `assignment` (
  `ass_id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(100) NOT NULL,
  `description` varchar(100) NOT NULL,
  `upload_document` text NOT NULL,
  `upload_document2` text NOT NULL,
  `upload_date` varchar(100) NOT NULL,
  `assessment_Id` varchar(100) NOT NULL,
  `student_email` varchar(100) NOT NULL,
  PRIMARY KEY (`ass_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Server:** 127.0.0.1
- Database:** student
- Table:** assignment
- Table Structure View:** The "Table structure" tab is selected.
- Columns:** The table has 8 columns, each with a checkbox for selection.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ass_id	int(11)			No	AUTO_INCREMENT		Change Drop Primary Unique Index Spatial More
2	title	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More
3	description	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More
4	upload_document	text			No	None		Change Drop Primary Unique Index Spatial More
5	upload_document2	text			No	None		Change Drop Primary Unique Index Spatial More
6	upload_date	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More
7	assessment_Id	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More
8	student_email	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More

- Actions:** Below the table structure, there are buttons for "Check all", "With selected:", "Browse", "Change", "Drop", "Primary", "Unique", "Index", "Add to central columns", "Remove from central columns", "Print view", "Propose table structure", "Track table", "Move columns", and "Improve table structure".

Figure: Screenshot for Assignment table

Department:

```
CREATE TABLE IF NOT EXISTS `department` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `department_name` varchar(100) NOT NULL,
  `facultyID` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```

The screenshot shows the 'Table structure' tab for the 'department' table. The table has three columns: 'id', 'department_name', and 'facultyID'. The 'id' column is defined as int(11) with AUTO_INCREMENT, primary key, and unique index. The 'department_name' column is defined as varchar(100) with primary key and unique index. The 'facultyID' column is defined as int(11) with primary key and unique index.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	department_name	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More
3	facultyID	int(11)			No	None		Change Drop Primary Unique Index Spatial More

Figure: Screenshot for Department table

Faculty:

```
CREATE TABLE IF NOT EXISTS `faculty` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `facultyName` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

The screenshot shows the 'Table structure' tab for the 'faculty' table. The table has two columns: 'id' and 'facultyName'. The 'id' column is defined as int(11) with AUTO_INCREMENT, primary key, and unique index. The 'facultyName' column is defined as varchar(100) with primary key and unique index.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
2	facultyName	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

Figure: Screenshot for Faculty table

Eclaim:

```
CREATE TABLE IF NOT EXISTS `eclaim` (
  `ec_id` int(20) NOT NULL AUTO_INCREMENT,
  `title` varchar(500) DEFAULT NULL,
  `description` varchar(1000) DEFAULT NULL,
  `upload_document` varchar(500) DEFAULT NULL,
  `upload_document2` varchar(500) DEFAULT NULL,
  `upload_document3` varchar(500) DEFAULT NULL,
  `upload_document4` varchar(500) DEFAULT NULL,
  `upload_date` date DEFAULT NULL,
  `assessment_Id` int(20) NOT NULL,
  `claimStatus` varchar(500) DEFAULT NULL,
  `processing_status` varchar(100) NOT NULL,
  `student_email` varchar(100) NOT NULL,
  PRIMARY KEY (`ec_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=36 ;
```

The screenshot shows the MySQL Workbench interface with the 'Table structure' tab selected for the 'eclaim' table. The table has 12 columns:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ec_id	int(20)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	title	varchar(500)			Yes	NULL		Change Drop Primary Unique Index Spatial More
3	description	varchar(1000)			Yes	NULL		Change Drop Primary Unique Index Spatial More
4	upload_document	varchar(500)			Yes	NULL		Change Drop Primary Unique Index Spatial More
5	upload_document2	varchar(500)			Yes	NULL		Change Drop Primary Unique Index Spatial More
6	upload_document3	varchar(500)			Yes	NULL		Change Drop Primary Unique Index Spatial More
7	upload_document4	varchar(500)			Yes	NULL		Change Drop Primary Unique Index Spatial More
8	upload_date	date			Yes	NULL		Change Drop Primary Unique Index Spatial More
9	assessment_Id	int(20)			No	None		Change Drop Primary Unique Index Spatial More
10	claimStatus	varchar(500)			Yes	NULL		Change Drop Primary Unique Index Spatial More
11	processing_status	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More
12	student_email	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More

Below the table structure, there are several navigation and action buttons:

- Check all / With selected: Browse Change Drop Primary Unique Index Add to central columns
- Remove from central columns
- Print view Propose table structure Track table Move columns Improve table structure
- Activate Windows
Go to Settings to activate Windows.

Figure: Screenshot for Eclaim table

Management:

```
CREATE TABLE IF NOT EXISTS `managementinfo` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `role` varchar(100) NOT NULL,
  `facultyName` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

The screenshot shows the MySQL Workbench interface with the 'managementinfo' table selected. The table has five columns: id, email, password, role, and facultyName. Each column is of type varchar(100) and is not null. The id column is set to auto-increment. Primary keys are defined for each column. There are also unique and spatial indexes.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
2	email	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	password	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
4	role	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
5	facultyName	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

Check all With selected: Browse Change Drop Primary Unique Index Add to central columns Remove from central columns

Print view Propose table structure Track table Move columns Improve table structure

Figure: Screenshot for Management table

Password_retrieve:

```
CREATE TABLE IF NOT EXISTS `password_retrieve` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(100) NOT NULL,
  `confirmation` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

The screenshot shows the MySQL Workbench interface with the 'password_retrieve' table selected. The table has three columns: id, email, and confirmation. The id column is of type int(11) and is set to auto-increment. Primary keys are defined for each column. There are also unique and spatial indexes.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
2	email	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More
3	confirmation	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

Check all With selected: Browse Change Drop Primary Unique Index Add to central columns Remove from central columns

Print view Propose table structure Track table Move columns Improve table structure

Figure: Screenshot for Password_retrieve table

Student:

```
CREATE TABLE IF NOT EXISTS `studentinformation` (
  `st_id` int(20) NOT NULL AUTO_INCREMENT,
  `st_FirstName` varchar(100) DEFAULT NULL,
  `st_Lastname` varchar(100) DEFAULT NULL,
  `st_dob` varchar(50) DEFAULT NULL,
  `st_email` varchar(100) DEFAULT NULL,
  `st_phon` varchar(30) DEFAULT NULL,
  `st_address` varchar(200) DEFAULT NULL,
  `st_photo` varchar(200) DEFAULT NULL,
  `st_password` varchar(30) DEFAULT NULL,
  `batchNo` varchar(30) DEFAULT NULL,
  `session` varchar(100) DEFAULT NULL,
  `Faculty` varchar(100) DEFAULT NULL,
  `department` varchar(100) NOT NULL,
  PRIMARY KEY (`st_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

The screenshot shows the 'Table structure' tab in MySQL Workbench for the 'studentinformation' table. The table has 13 columns:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	st_id	int(20)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext More
2	st_FirstName	varchar(100)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
3	st_Lastname	varchar(100)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
4	st_dob	varchar(50)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
5	st_email	varchar(100)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
6	st_phon	varchar(30)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
7	st_address	varchar(200)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
8	st_photo	varchar(200)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
9	st_password	varchar(30)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
10	batchNo	varchar(30)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
11	session	varchar(100)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
12	Faculty	varchar(100)			Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext More
13	department	varchar(100)			No	None		Change Drop Primary Unique Index Spatial Fulltext More

Below the table structure, there are buttons for 'Check all', 'With selected:', 'Print view', 'Propose table structure', 'Track table', 'Console', 'Activate Windows', and 'Go to Settings to activate Windows'.

Figure: Screenshot for Student table

1.5 Retrieving Data with joining:

In my database, I properly create relationship so that we can access different tables' data from one table and we create it successfully. Below I give some example of retrieving data:

- I create a joining query to student table with faculty and department table. I created relationships between these tables successfully. Therefore, Now I can retrieve data from those tables easily by using inner join.

Query - 1:

```
SELECT st_id AS StudentID, st_LastName AS Lastname, st_email AS Email, faculty.facultyNa  
me AS Faculty, department.department_name AS Department  
FROM `studentinformation`  
INNER JOIN faculty ON faculty.id = studentinformation.faculty  
INNER JOIN department ON department.id = studentinformation.department  
ORDER BY st_id  
LIMIT 0 , 30
```

StudentID	Lastname	Email	Faculty	Department
1	Shiplu	shiplu@gmail.com	Science	IT
2	Rumi	rumi@gmail.com	Science	IT
3	nazmul	nazmul@gmail.com	Science	IT

Figure: Showing student information with faculty & department

- I also create an inner join query to join assessment table with faculty and department table. Here, I also become successful in this query because of my successful relationship between the table.

Query - 2:

```

SELECT 30ssessment.id AS AssessmentID, ass_name AS Name, sessions, batch_number AS
Batch, documents AS Document,faculty.facultyName AS Faculty, department.department_nam
e AS Department
FROM `30ssessment`
INNER JOIN faculty ON faculty.id = 30ssessment.facultyName
INNER JOIN department ON department.id = 30ssessment.departmentName
ORDER BY 30ssessment.id
LIMIT 0 , 30

```

AssessmentID	Name	sessions	Batch	Document	Faculty	Department
19	Wright paragraph23	2017	35	claim_problem/balance.png	Science	IT
20	Gaming Football	2017	35	claim_problem/1555543_569591643145569_499356190508...	Arts	ETE
21	Gaming Football	2017	35	claim_problem/1460405302-porch0516.jpg	Science	ETE
22	Gaming Football	2017	35	claim_problem/84020749833147.58bfec86724fe.jpg	Science	IT
23	Wright Letter 2	2017	35	claim_problem/ae009e49608233.58b9fe7153ba5.jpg	Arts	Law
24	New Assessment	2017	35	claim_problem/d.png	Science	IT
25	dg	dg	dg	claim_problem/banana.png	Science	IT
26	Wright paragraph	2017	35	claim_problem/bamboo.png	Humanities	ETE
27	cgg	4444	34	claim_problem/balance.png	Arts	ETE
28	New Assessment	2017	35	claim_problem/2K1Q7U.pdf	Arts	BBA
29	New Assessment22	2017	35	claim_problem/2K1Q7U.pdf	Science	English
30	New Assessment	2017	35	claim_problem/EBEK APPLICATION FORM MANJURUL.pdf	Science	BBA
31				claim_problem/6H4U3P.pdf	Science	IT
32	EWSD	2017	35	claim_problem/Capture.PNG	Science	IT
33	DW	2017	34	claim_problem/Capture.PNG	Science	IT
34	DW	2017	35	claim_problem/Capture.PNG	Science	IT

Figure: Assessment information with faculty & department

1.6 Security Measures in the Database:

I make my database fully secured so that non-authenticated person cannot access our site data. For this reason, I use some techniques like:

- ✓ My management table is quite sensible so I did not use password as a plain text. For password security, I make the password encrypted by using MD5 technology.

+ Options	← →	▼	id	email	password	role	facultyName
	<input type="checkbox"/> Edit Copy	1	tar711@gmail.com	202cb962ac59075b964b07152d234b70		ec_manager	
	<input type="checkbox"/> Edit Copy	2	tarikul711@gmail.com	202cb962ac59075b964b07152d234b70		ec_coordinator	1
	<input type="checkbox"/> Edit Copy	3	tonikul711@gmail.com	202cb962ac59075b964b07152d234b70		admin	

Check All With selected:

Figure: Encrypted password in management table

- ✓ I also use SQL Injection so that no one can easily hack our database or make database weak.
- ✓ I also give data validation by given proper data type for the attributes. For date, number and other fields I use the proper data type.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(20)			No	None	AUTO_INCREMENT	More
2	ass_name	varchar(200)	latin1_swedish_ci		Yes	NULL		More
3	facultyName	varchar(200)	latin1_swedish_ci		Yes	NULL		More
4	batch_number	int(100)			No	None		More
5	sessions	varchar(100)	latin1_swedish_ci		No	None		More
6	dates	date			Yes	NULL		More
7	duDate	date			Yes	NULL		More
8	lastSubDate	date			Yes	NULL		More
9	documents	text	latin1_swedish_ci		No	None		More
10	departmentName	varchar(100)	latin1_swedish_ci		No	None		More

Check All With selected:

Figure: The structure of assessment table

- ✓ I also create role based management system. No one can enter the different role person table. The system will identify the role automatically by their email address.

+ Options

			id	email	password	role	facultyName
<input type="checkbox"/>	  	1	tar711@gmail.com	202cb962ac59075b964b07152d234b70		ec_manager	
<input type="checkbox"/>	  	2	tanikul711@gmail.com	202cb962ac59075b964b07152d234b70		ec_coordinator	1
<input type="checkbox"/>	  	3	tonikul711@gmail.com	202cb962ac59075b964b07152d234b70		admin	

Check All With selected:  Change  Delete  Export

Figure: Management table with different role people