**ONLINE LEARNING MANAGEMENT SYSTEM**

**Table of Contents**

1. Introduction
2. System Requirements
3. ER diagram
4. Schema Diagram and Normalization
5. Creation and insertion of tables
6. Queries Used
7. Data Retrieval – REST API’s
8. UI Screens
9. Execution Process

## **Introduction**

Online Learning Management system is a platform which helps the instructors create their own set of private websites were student can login and access the content the courses, participate in the quizzes and get their grades. Also, every student can discuss their own topic with the fellow students. Instructors can conduct surveys and workshops. The OLMS can be made more customizable according to the user’s needs



Access the Database



Function Calls

Libraries and API’s

User Interface

There are other features such as providing feedback for the course content, Also the exam and assignments details can be provided in the OLMS. This is small tool which can b used by the set of friends where one person can be admin and assign tasks to the fellow student.

1. **System Requirements**

Online Learning Management System is primarily developed in MAC operating system using MySQL for the backend. For the interface between the backend and frontend we use Springboot. User Interface is developed using the React JS. Basic Requirements to setup this Online Learning Management System is

**Hardware**

* Disk space: 200MB for the code, 2 GB is probably a realistic minimum.
* Processor: 1 GHz (min), 2GHz dual core or more recommended.
* Memory: 512MB (min), 1GB or more is recommended. 8GB plus is likely on a large production server

**Software**

Mentioned below are the minimum supported versions. The recommend will be the latest versions of the software’s.

* Spring Boot: 2.6.5
* ReactJS: These are just the minimum supported versions. We recommend keeping all of your software up to date.

**Database requirements**

For the development purpose we used My

|  |  |  |
| --- | --- | --- |
| **Database** | **Minimum version** | **Recommended** |
| MySQL | 5.5.31 | Latest |

Table 2.1 Database requirement

## **ER Diagram**

Diagram, schematic

Description automatically generated

**ASSUMPTIONS**

1) Admin, student, instructor and support have Account type.

2) Student and instructor have department.

3) Student post forum and instructor replies to the posted forum.

4) One department have many students and many instructors.

5) Instructor conducts many quizzes and students gets marks to the conducted quiz.

6) Each quiz has grades.

7) Instructor conducts multiple workshops, and zero or more students participate in the workshop.

8) Each student has his own portfolio.

9) Instructor conducts zero or many surveys and students participate in it.

10) Each semester has many subjects.

11) Admin gives the announcements.

12) Each Course have content and student can browse it.

13) Support provides contact information.

14) One instructor can provide much feedbacks.

15) One instructor can conduct many workshops.

16) Survey is considered as general survey which can be taken by anyone.

17) Announcements are posted by admin

1. **Schema Diagram and Normalization**

Graphical user interface, application, table

Description automatically generated

## **Creation and insertion of tables**

* 1. **Student**: The major and most element of the Online Learning Management is Student. The table has 5 attributes
  + student\_id – Unique identification for each student. This attribute will be the primary key for this table
  + student\_name – This column stores the names of each student
  + student\_phone\_number – this column stores the phone number of the student
  + student\_email – To the store the email address of each student
  + student\_address – To store the student address of each student
* Below is the query to create the table for student

create table student (student\_id INT, student\_name varchar(50) not null, student\_phone\_number varchar(20),student\_email varchar(20), student\_address varchar(500), primary key (Student\_id));

* Below is the statement to insert value into the student table

Insert into Student

(Student\_id, Student\_name, Student\_phone\_number, Student\_email, Student\_address)

VALUES

(12345, "Akshay", "940-781-4201","Akshay@GMAIL.COM","621 BERNARD STREET"),

(67849, "Akhil", "840-611-4961","Akhil@GMAIL.COM","618 Venue"),

(34789, "Koushik", "730-891-8761","Koushik@GMAIL.COM","11 West mulberry"),

(90456, "Jaidev", "890-130-0531","Jaidev@GMAIL.COM","22 The warton drive"),

(81456, "Ramu", "230-298-5211","Ramu@GMAIL.COM","301 tarton enclave");

* 1. **Instructor**: One more major and most element of the Online Learning Management is Instructor. The table has 5 attributes
  + instructor\_id – Unique identification for each Instructor. This attribute will be the primary key for this table
  + instructor\_name – This column stores name of each Instructor
  + instructor\_phone\_number – this column stores the phone number of the instructor
  + instructor\_email – To the store the email address of each instructor
  + instuctor\_experience – To store how many years of experience each instructor has.
* Below is the query to create the table for instructor

create table instructor (instructor\_id INT, instructor\_name varchar (50) not null, instructor\_phone\_number varchar (20), instructor\_email varchar (30), instuctor\_experience INT, Course\_id INT, primary key (Instructor\_id));

* Below is the statement to insert values into the instructor table

Insert into Instructor

(Instructor\_id, Instructor\_name, Instructor\_phone\_number, Instructor\_email, Instuctor\_Experience)

VALUES

(11223, "Mark", "910-781-4001","Mark@GMAIL.COM",19),

(69812, "Zing", "830-616-4871","Zing@GMAIL.COM",8),

(31768, "Wei", "760-991-8561","Wei@GMAIL.COM",9),

(98876, "Lung", "880-190-6541","Lung@GMAIL.COM",6),

(80311, "Mufasa", "270-297-8911","Mufasa@GMAIL.COM",15);

* 1. **Course**: Student takes courses and instructor teach teaches courses. We need to have the table to store the course info. The table has 3 attributes
  + course\_id – unique identification number for each course. This attribute will be the primary key for this table
  + course\_name – This column stores the courses name.
  + course\_duration – Each course has the specific period, which is stored in this column.
* Below is the query to create the table for course

create table course (course\_id INT, course\_name varchar(50) not null, course\_duration varchar(20),primary key (Course\_id));

* Below is the statement to insert value into the course table

Insert into Course

(course\_id, course\_name, course\_duration)

VALUES

(67, "Data\_Base","5 week"),

(78, "Machine\_Learning","8-week"),

(12, "Computer\_Algo","11-week"),

(11, "Graph\_theory","9-week"),

(42, "Software\_Engineering","10-week");

* 1. **student\_course**: We need to know which student takes what course. To get this normalization is applied. After normalization it has the following attributes
  + student\_course\_id – unique identification number which tells the student’s course. This attribute will be the primary key for this table
  + course\_id – unique identification number for each course
  + student\_id – unique identification for each student.
* Below is the query to create the table for student\_course

create table student\_course ( student\_course\_id int AUTO\_INCREMENT,student\_id int,course\_id int, primary key (student\_course\_id), FOREIGN KEY (student\_id) REFERENCES student(student\_id), FOREIGN KEY (course\_id) REFERENCES course(course\_id));

* Below is the statement to insert value into the student\_course table

insert into student\_course(student\_id,course\_id) values

(12345,67),(12345,67),(67849,78),(67849,12),(12345,12),(12345,11),(12345,42),

(90456,11);

* 1. **department**: This table is used to store the department info. which will be helpful to know what courses are present in which department. The table has following attributes After normalization it has the following attributes
  + department\_id – unique identification given to each. This attribute will be the primary key for this table
  + department\_name – This column stores the department name
  + instructor\_id – Every department has instructors. Those instructor id’s are stored in this columninstructors unique identification for each student.
* Below is the query to create the table for department table

create table department (department\_id INT, department\_name varchar(50) not null,instructor\_id INT, primary key (department\_id), FOREIGN KEY (instructor\_id) REFERENCES instructor(Instructor\_id));

* Below is the statement to insert value into the department table

Insert into Department(Department\_id, Department\_name,Instructor\_id )

VALUES(1,"AI",11223),(2,"IS",69812), (3,"CS",31768), (4,"MIS",98876),

(5,"DS",80311);

* 1. **Semester**: This table is used to store the semester info. which will be helpful to know currently which student is enrolled into which semester. The table has following attributes
  + semester\_id – unique identification given to each semester. This attribute will be the primary key for this table
  + semester\_no – This column stores the semester number.
  + Semester\_name – this column stores the semester name.
* Below is the query to create the table for semester table

create table semester (student\_id int,semster\_id int AUTO\_INCREMENT,semester\_noINT, semester\_name varchar(50) not null, primary key (semster\_id));

);

* Below is the statement to insert value into the semester table

Insert into Semester

(student\_id,semester\_No, Semester\_name )

VALUES

(12345,1,"Fall"),

(67849,2,"Spring"),

(34789,3,"Summer"),

(81456,4,"Winter");

* 1. **fees**: This table is used to store the fees info of each student. The table has following attributes
  + fee\_id – unique identification given to each fee payment. This attribute will be the primary key for this table
  + semester\_no – This column stores the semester number. Tells the fee for each semester
  + Amount – This column tells the amount to be paid.
  + Statsu – This column tells the status of the fee whether it is paid for not.
* Below is the query to create the table for fees table

create table fees(fee\_id int AUTO\_INCREMENT,Semester\_No INT,Amount float not null, fee\_status Boolean,primary key (fee\_id));

* Below is the statement to insert value into the fees table

Insert into fees

(Student\_id,Semester\_No, Amount,fee\_status )

VALUES

(12345, 1, 5674,0),

(67849, 2, 3412,0),

(34789, 3, 5612,0),

(90456, 4, 0,1),

(81456, 1, 100,0);

* 1. **portfolio**: every student has a profile. This table is used to store the details of each student. The table has following attributes
  + portfolio\_id – unique identification given to each profile. This attribute will be the primary key for this table
  + Profile\_name – name of the profile.
  + Profile\_description – This column stores the description of each profile.
  + Student\_id : This column stores the student id’s
* Below is the query to create the table for portfolio table

create table portfolio (portfolio\_id int AUTO\_INCREMENT, student\_id INT, profile\_name varchar (20), profile\_description varchar (500) not null, primary key (portfolio\_id) );

* Below is the statement to insert value into the portfolio table

Insert into Portfolio

(Student\_id,Profile\_name,Profile\_description )

VALUES

(12345, "Aksh@y", "I have 6 years of experience in Database"),

(67849, "Akhil$$$", "I have 8 years of experience in the Data engineer and worked as team lead"),

(34789, "Koushik^^", "I am experitised in the saleforce and have my own startup"),

(90456, "JaidevGIlla","I have recently passed out from the college, I have conducted carpedium fest"),

(81456, "Ramachandr@@@","Well known subjects are SQL and DBMS");

* 1. **Account\_Type**: This table is used to store the account details of every user whether it might be student or instructor. The table has following attributes
  + account\_id – Unique identification given to each user. This attribute will be the primary key for this table
  + account\_password – This column stores the password for each user.
  + Account\_email – This column stores the emails of all the users
* Below is the query to create the table for Account\_Type table

create table account\_Type(account\_id INT,account\_Password varchar(20), account\_email varchar(20),primary key (account\_id));

* Below is the statement to insert value into the account\_type table

Insert into account\_type

(account\_id, account\_Password, account\_email)

VALUES

(12345, "aks","Akshay@GMAIL.COM"),

(67849, "akh","Akhil@GMAIL.COM"),

(34789, "kou","Koushik@GMAIL.COM"),

(90456, "jai","Jaidev@GMAIL.COM"),

(81456, "ram","Ramu@GMAIL.COM"),

(11223, "mar","Mark@GMAIL.COM"),

(69812, "zin","Zing@GMAIL.COM"),

(31768, "wei","Wei@GMAIL.COM"),

(98876, "lun","Lung@GMAIL.COM"),

(80311, "muf","Mufasa@GMAIL.COM");

* 1. **assignment**: This table is used to store the assignment details of each student. The table has following attributes
  + student\_id – unique identification given to each student. This attribute will be the primary key for this table
  + assignment\_title – This column stores the titles of each assignment.
  + Start\_date – This column tells the start\_date of the assignment.
  + Due\_Date – This column tells the end\_Date of the assignment.
* Below is the query to create the table for assignment table

create table assignment (student\_id INT,assignment\_id INT AUTO\_INCREMENT, assignment\_title varchar(200), start\_date date,due\_date date, primary key (Assignment\_id));

* Below is the statement to insert value into the assignment table

insert into assignment (student\_id,

assignment\_title,

start\_date,

due\_date) values

('12345','Data base assignment 1','2022-04-01','2022-04-10'),

('67849','Machine\_Learning','2022-03-01','2022-03-10'),

('12345','Software Engineering Assignment 1','2022-03-01','2022-03-10'),

('90456','Graph\_theory Assignment','2022-01-23','2022-02-05');

* 1. **workshop**: Workshop gives the practical exposure to the student. The details of the workshop conducted are stored in this table. The table has following attributes
  + workshop\_id – unique identification given to each workshop. This attribute will be the primary key for this table
  + workshop\_subject – This column stores the subject on which the workshop is conducted.
  + Workshop\_date – This column stores the date on which the workshop is conducted
  + Instructor\_id– This column stores the instructor\_id which tells the instructors who conducted the workshop
* Below is the query to create the table for workshop table

create table workshop(workshop\_id INT,workshop\_subject varchar(50), workshop\_date date,instructor\_id int, primary key (workshop\_id));

* Below is the statement to insert value into the workshop table

insert into workshop(workshop\_id,workshop\_subject,workshop\_date)

values(1,'Graph Theory Topic','2022-02-03'),

(2,'IOT Topic','2022-02-19');

* 1. **grade**: This table is used to store the grades of each student. The table has following attributes
  + grade\_id – unique identification given to each grade. This attribute will be the primary key for this table
  + exam\_no – This column stores the exam number.
  + Marks – This column tells the marks of the student.
  + grade – This column stores the grades of the student
* Below is the query to create the table for grade table

create table grade(grade\_id int, exam\_no INT,marks INT, grade varchar(10),primary key (grade\_id),FOREIGN KEY (exam\_no) REFERENCES exam(exam\_no));

* Below is the statement to insert value into the grades table

insert into grade(grade\_id,exam\_no,grade,marks) values(1,1,'A',98),

(2,2,'B',82),(3,3,'A',95),(4,4,'B',80)

* 1. **exam**: This table is used to store the exam details of each student. The table has following attributes
  + student\_id – unique identification given to each student. This attribute will be the primary key for this table
  + exam\_no – This column stores the exam number.
  + Exam\_name – This column tells the name of the exam
  + Exam\_Date – This column tells the date on which the exam is conducted.
* Below is the query to create the table for exam table

create table exam(student\_id INT, exam\_no INT, exam\_name varchar(30), exam\_date date, primary key (exam\_no));

* Below is the statement to insert value into the exam table

insert into exam(exam\_no,exam\_name,exam\_date) values

(1,'graph Theory exam -1','2022-03-10')

(12345,2,'graph Theory exam final','2022-04-10')

(67849,3,'Machine Learning exam-1','2022-03-01'),

(67849,4,'Machine Learning Final','2022-05-02')

* 1. **feedback**: This table is used to store the feedback given by each student for the course. The table has following attributes
  + student\_id – unique identification given to each student.
  + Feedback\_id – unique id for each feedback provided. This attribute will be the primary key for this table
  + Feedback\_text – column to store the text provided for each feedback
* Below is the query to create the table for feedback table

create table feedback(feedback\_course\_id INT,feedback\_id INT AUTO\_INCREMENT, feedback\_text varchar(300),primary key (feedback\_id),FOREIGN KEY (student\_id) REFERENCES student(student\_id));

* Below is the statement to insert value into the feedback table

insert into feedback(

feedback\_course\_id,

feedback\_text) values('11','Fundamentals of data base is good subjects help to understand mysql relations’,

query fetching and all concepts related to design the application'),

('67849','78','ML subject is very good ');

* 1. **content**: This table is used to store the content for each course. The table has following attributes
  + content\_id – unique identification given to each content. This attribute is primary for this table
  + content\_type – this column stores the type of the content, that is whether it is theory or practical
  + content\_description – This column stores the content description
* Below is the query to create the table for content table

create table content( content\_id INT,content\_type varchar(50), content\_description varchar(50),course\_id int,primary key (content\_id));

* Below is the statement to insert value into the content table

insert into content(content\_id,content\_type,content\_description) values(1,'Theory','chapter 1'),

(2,'Theory','chapter 2',67),(3,'Theory','chapter 3');

* 1. **survey**: In general, survey is conducted to the student’s response on various topics. The table has following attributes
  + survey\_id – unique identification given to each survey conducted.
  + Survey\_topic – This column tells the topic on which the survey is conducted
  + Survey\_Active – This column tells the status of the survey whether it is currently active or not
  + instructors\_id – unique id od the instructor which tells the name of instructor who conducted the survey
* Below is the query to create the table for survey table

create table survey(survey\_id INT, survey\_topic varchar(50), survey\_active boolean, instructor\_id int, primary key (survey\_id));

* Below is the statement to insert value into the survey table

insert into survey(survey\_id,survey\_topic,survey\_active)values(1,'About the course',true),

(2,'About the course',true);

* 1. **forum**: we have a feature where students can discuss with the fellow students or instructor. To store the data discussed we use forum table.
  + forum\_id – unique identification given to each thread in the discussion. This attribute will be the primary key for this table
  + student\_id – this tells which student has posted the discussion
  + forum\_text – The text what student keeps is stored in this table.
* Below is the query to create the table for forum table

create table forum( forum\_id int, student\_id int, forum\_text varchar(150), primary key (forum\_id));

* Below is the statement to insert value into the forum table

insert into forum(forum\_id,forum\_text)

values(1,'who is best is instructor in AI?')

,(2,'Which semester is Best enrolling the Algorithms subject?');subject?');

* 1. **Master\_account**: This stable is used to store the type of account, that whether it is student’s account or the instructor’s account.
  + master\_id – unique identification given to each of the account type. This attribute will be the primary key for this table
  + account\_type – This column tells the type of the account if it 1 then student, if it is 2 then it is instructor.
  + role\_name – The role name tells whether it is student’s or instructors.
* Below is the query to create the table for master\_account table

create table master\_account( master\_id INT, account\_type int, role\_name varchar(50) not null, primary key (master\_id));

* Below is the statement to insert value into the master\_account table

insert into master\_account(master\_id,account\_type,role\_name)values

(1,1,'student'),(2,2,'instructor');

* 1. **support**: This table is used to store the contact details of the support staff such as admin and help desk. The table has the following attributes
  + support\_id – unique identification given to each support provided. This attribute will be the primary key for this table
  + support\_type – This column stores the type of support staff whether it is admin or helpdesk
  + contact\_details – In this column we store the contact details of the support staff
* Below is the query to create the support table

create table support (support\_id INT, support\_type varchar(50),contact\_details varchar(50), primary key (support\_id));

* Below is the statement to insert value into the support table

insert into support(support\_id,support\_type,contact\_details)

values(1,'Admin','9401231234'),(2,'Help Desk','9401223412');

* 1. **associate\_instructor**: This table is used to store the assistant details who work for the instructor. The table has 3 attributes
* associate\_instructor\_id – unique identification given to each. This attribute will be the primary key for this table
  + associate\_instructor\_name – This column stores the names of the associate instructor
  + Survey\_Active – This column tells the status of the survey whether it is currently active or not
  + instructor\_id – Every associate instructor works for an instructor. Those instructor ids are stored in this column.
* Below is the query to create the table for associate\_instructor table

create table associate\_instructor(associate\_instructor\_id INT,associate\_instructor\_name varchar(50),associate\_email varchar(50), instructor\_id int,primary key (associate\_instructor\_id));

* Below is the statement to insert value into the associate\_instructor table

insert into associate\_instructor(associate\_instructor\_id, associate\_instructor\_name,

associate\_email)

values(1,'ram','ram@unt.com'),(2,'sita','sita@unt.com'),

(3,'rakesh','rakesh@unt.com'),(4,'raju','raju@gmail.com');

**u. announcements**: This table is used to store accountments given by the admin to the students

* announcement\_id – unique identification given to each announcement. This attribute will be the primary key for this table
  + announcement\_text – This column stores text of announcement given.
  + announcement\_Active – This column tells whether the given announcement whether it is currently active or not
  + admin\_id – As the announcement are given by the admin. There id’s are stored in this column.
* Below is the query to create the table for announcment table

create table announcement (announcement\_id INT, announcement\_text varchar (50), announcement\_active boolean, admin\_id int,primary key (announcement\_id), FOREIGN KEY (admin\_id) REFERENCES admin(admin\_id));

* Below is the statement to insert value into the announcement table

insert into announcement (announcement\_id, announcement\_text,accouncement\_active,admin\_id) values(1,'Class will begin in 1 week of may',true,1), (2,'PLease pay your dues by 15th may',true,2);

**v. admin**: This table is used to store the admin details. The table has the following attributes

* admin\_id – unique identification given to each admin. This attribute will be the primary key for this table
  + admin\_name – This column stores the admin names.
* Below is the query to create the table for admin table

create table admin (admin\_id int,admin\_name varchar(50),primary key (admin\_id));

* Below is the statement to insert value into the admin table

insert into admin(admin\_id,admin\_name)values(1,'Ram'),(2,'raj')

1. **Queries Used**
2. Query to get all students info from all tables.

select asi.assignment\_title,asi.start\_date,asi.due\_date,co.course\_name,co.course\_duration,sem.semester\_no,

fees.amount,fees.fee\_status,ex.exam\_name,ex.exam\_date,foru.forum\_text,con.content\_description,

por.profile\_description

from student as sid

left join assignment asi on asi.student\_id=sid.student\_id

left join student\_course stucou on stucou.student\_id=sid.student\_id

left join course co on co.course\_id=stucou.student\_course\_id

left join semester sem on sem.student\_id=sid.student\_id

left join fees on fees.student\_id=sid.student\_id

left join feedback fed on fed.student\_id=sid.student\_id

left join exam ex on ex.student\_id=sid.student\_id

left join forum foru on foru.student\_id=sid.student\_id

left join content con on con.course\_id=co.course\_id

left join portfolio por on por.student\_id=sid.student\_id

where sid.student\_id=12345

group by asi.assignment\_title,asi.start\_date,asi.due\_date,co.course\_name,co.course\_duration,sem.semester\_no,

fees.amount,fees.fee\_status,ex.exam\_name,ex.exam\_date,foru.forum\_text,con.content\_description,

por.profile\_description;

select (stcou.course\_id) from student stu

left join student\_course stcou on stcou.student\_id=stu.student\_id

left join course cou on cou.course\_id=stcou.student\_course\_id

left join instructor ins on ins.course\_id=cou.course\_id

where stu.student\_id=12345

group by stcou.course\_id

1. Query to the courses which currently the student is enrolled into
2. Query to the associate instructore name by instructor ID

//get the associate instructor name by instrctor id

select instructor\_name,associate\_instructor\_name,associate\_email from instructor ins

left join associate\_instructor asins on asins.instructor\_id=ins.instructor\_id

where ins.instructor\_id=11223

1. As part of the normalization, created the master\_student\_exam table

create table master\_student\_exam(

student\_exam\_id int,

exam\_no int,

student\_id int,

primary key (student\_exam\_id),

FOREIGN KEY (exam\_no) REFERENCES exam(exam\_no),

FOREIGN KEY (student\_id) REFERENCES student(student\_id)

);

1. After normalization created the master\_survey\_instructor

create table master\_survey\_instructor(

sur\_ins\_id int,

instructor\_id int,

survey\_id int,

primary key (sur\_ins\_id),

FOREIGN KEY (instructor\_id) REFERENCES instructor(instructor\_id),

FOREIGN KEY (survey\_id) REFERENCES survey(survey\_id)

);

1. As part of the normalization created the master student feedback

create table master\_student\_feedback(

student\_feedback\_id int,

student\_id int,

feedback\_id int,

primary key (student\_feedback\_id),

FOREIGN KEY (student\_id) REFERENCES student(student\_id),

FOREIGN KEY (feedback\_id) REFERENCES feedback(feedback\_id)

);

1. After normalization, created the master student semster

create table master\_student\_semster(

student\_semester\_id int,

student\_id int,

semster\_id int,

primary key (student\_semester\_id),

FOREIGN KEY (student\_id) REFERENCES student(student\_id),

FOREIGN KEY (semster\_id) REFERENCES semester(semster\_id)

);

1. As part of normalization,created master student assignment table

create table master\_student\_assignment(

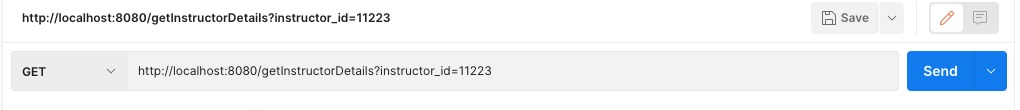
student\_assignment\_id int,

student\_id int,

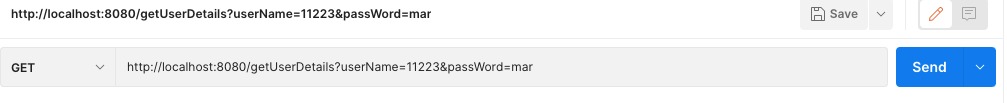
assignment\_id int);

# **Data Retrieval - REST API’s**

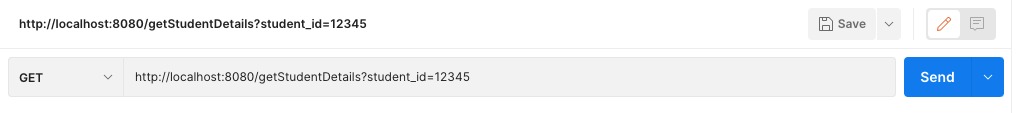
We use Rest APIs to fetch or update the data from the database.



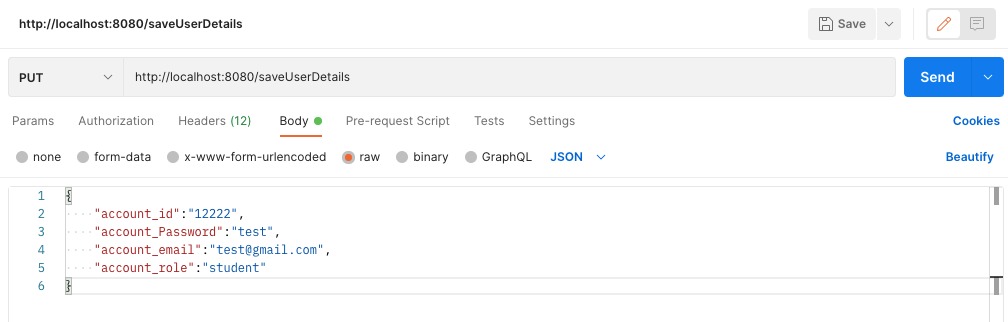
The above is the endpoint to retrieve the instructor details based on the instructor id.



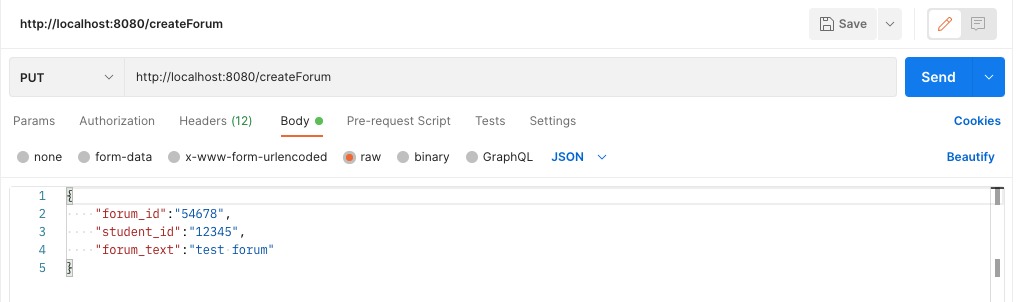
The above is the endpoint to get the account details based on the given username and password which is used for the login.



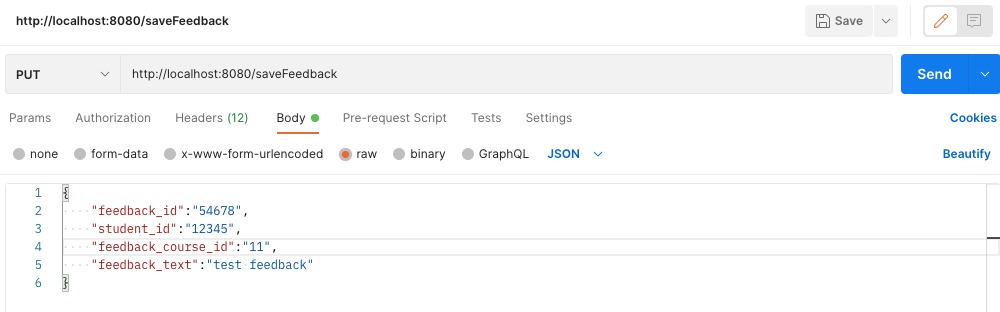
The above is the endpoint to retrieve the student details based on the student id.



The above endpoint is used to save the account details based on the given account type object as the request.

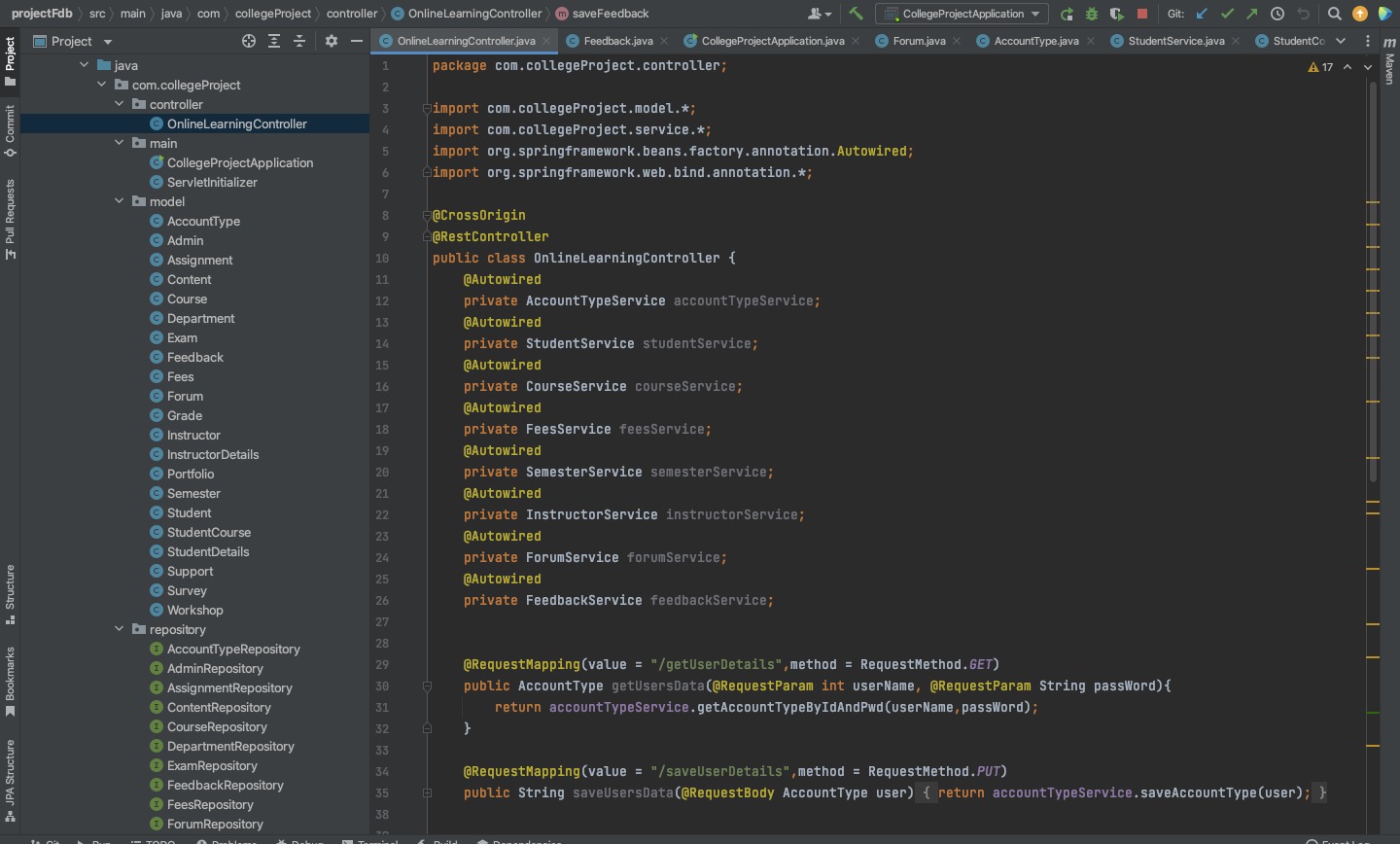


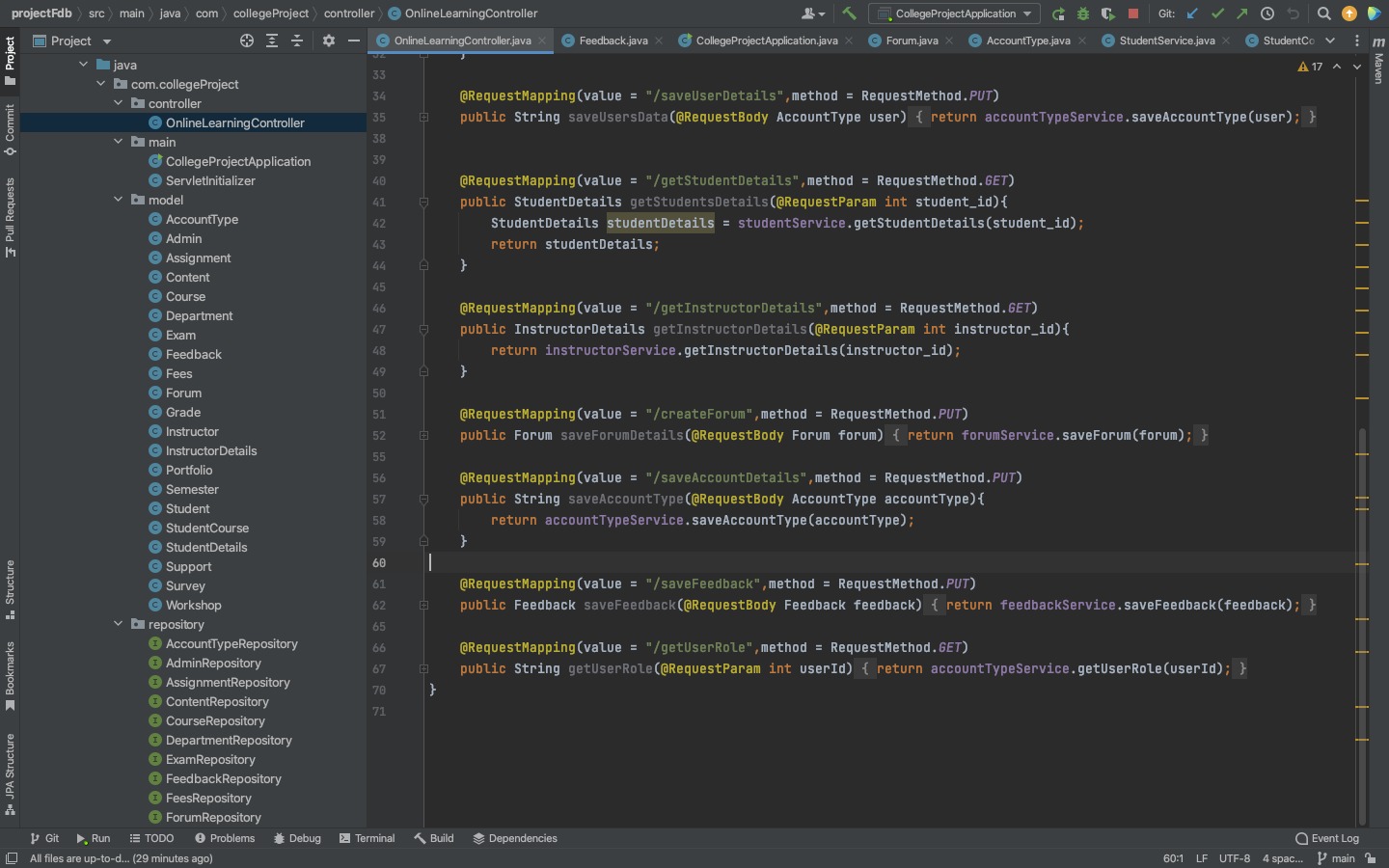
The above endpoint is used to publish any details in the forum.



The above endpoint is used to publish the feedback.

Below shown are the sample rest endpoints implementation in the code.





# 

# **UI Screens**

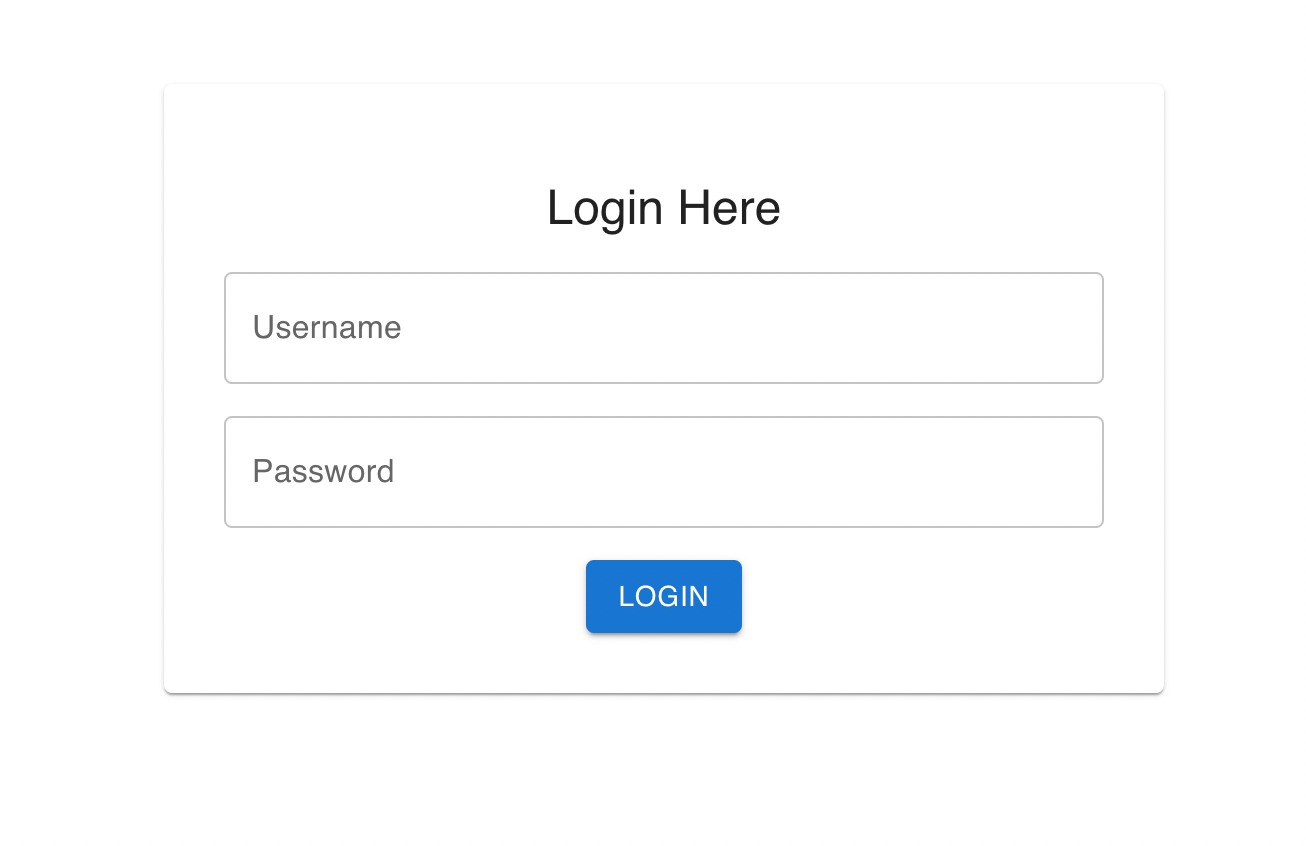


Figure 1

The above is the login page, which is common for both student or instructor. On entering the username and password in the backend it is verified whether it belongs to the student or instructor. Based on that it redirects to the next page

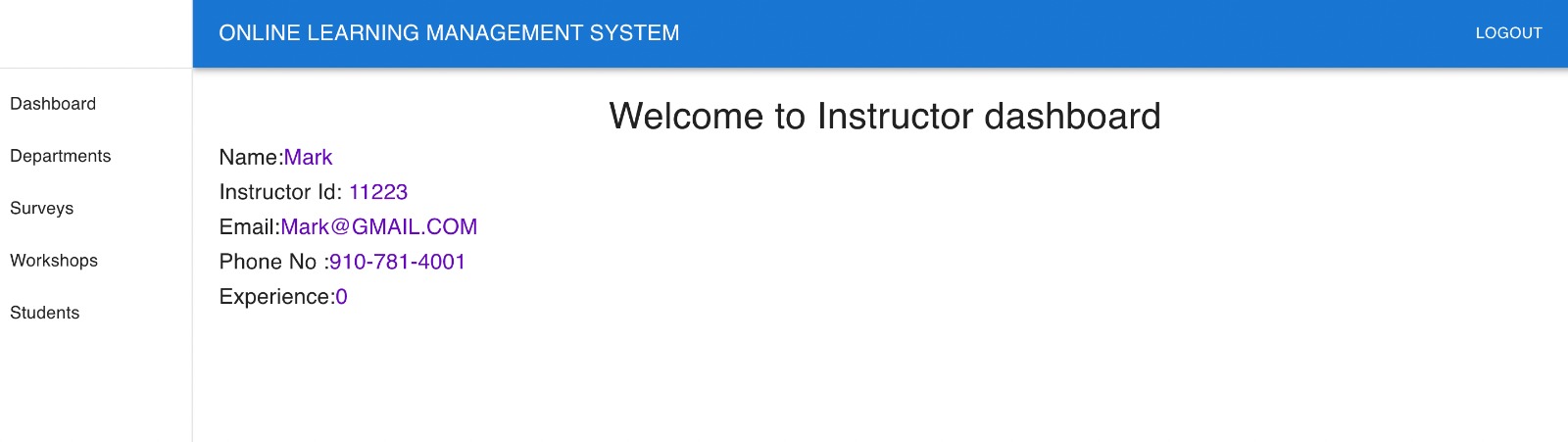


Figure 2

If the credentials entered in the page 1 is of instructor’s based on the username entered the following screen appears in which the side pane has the various field of options

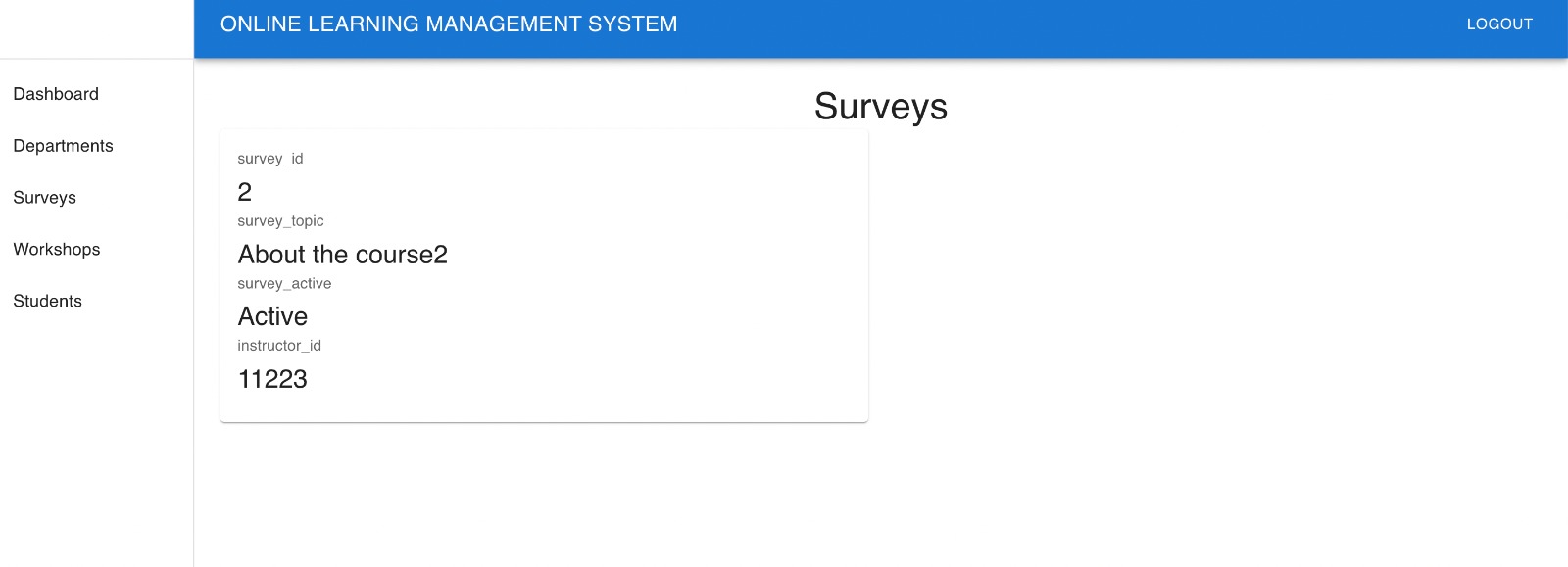


Figure 3

On clicking the surveys tab, the above figure appears mentioning the survey\_topic and the status of survey.

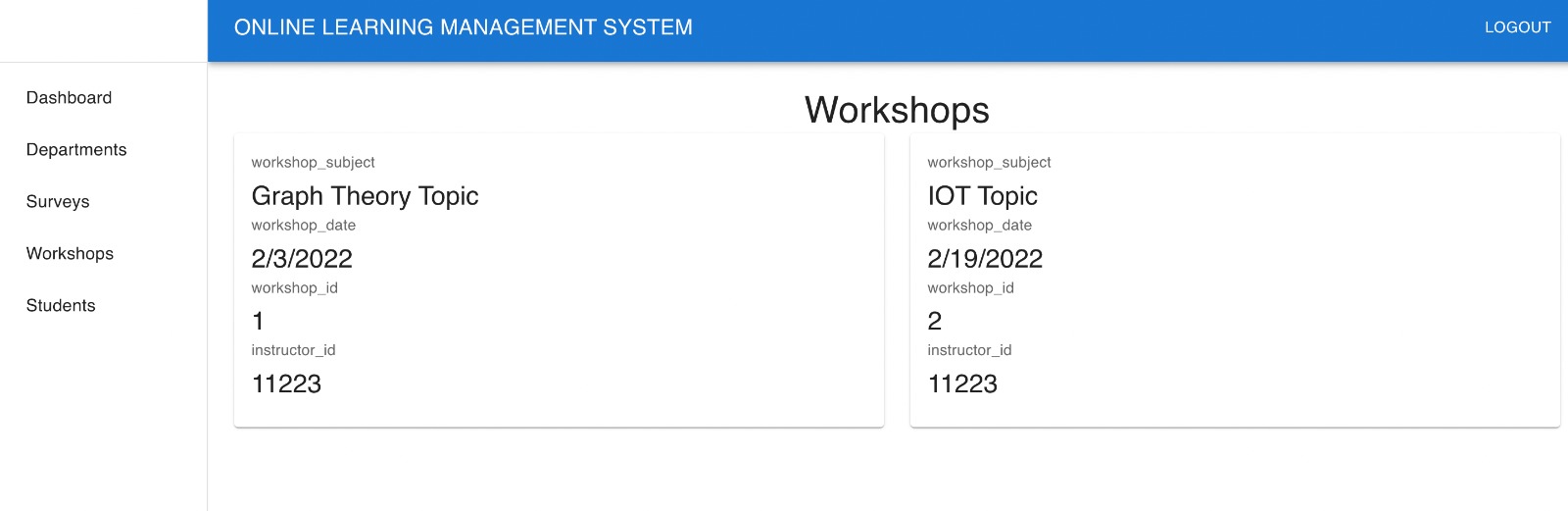


Figure 4

On clicking the workshops, the above figure appears which gives the details of workshop conducted and the dates of it.

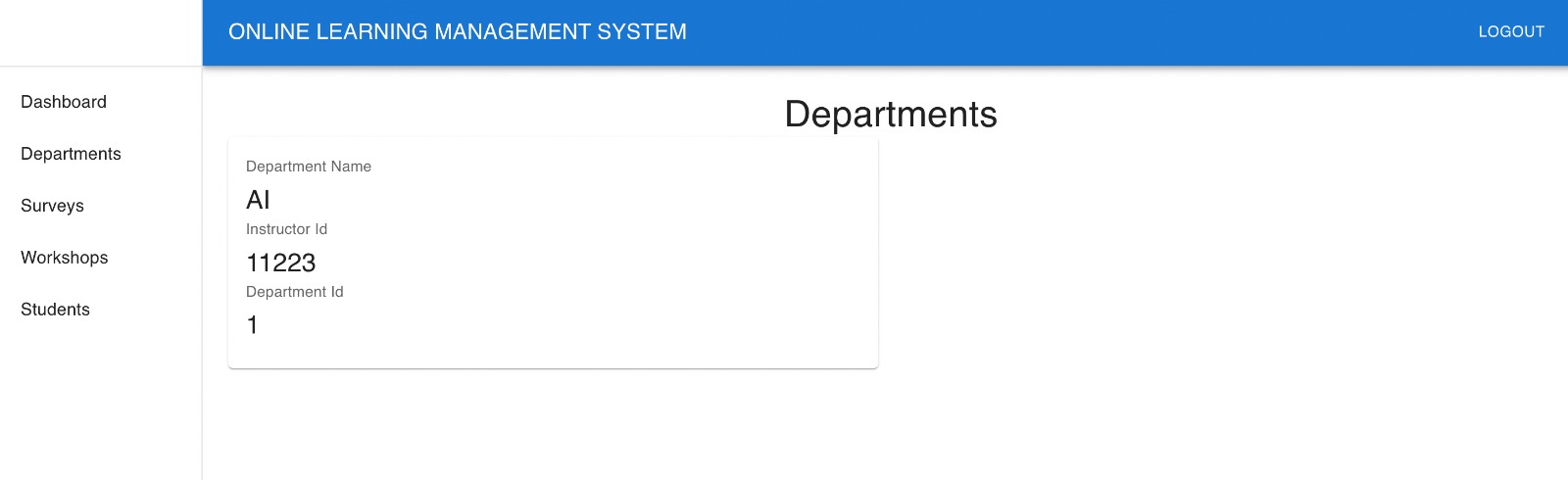


Figure 5

On clicking the department tab, it gives the details of the department which the professor belongs to.

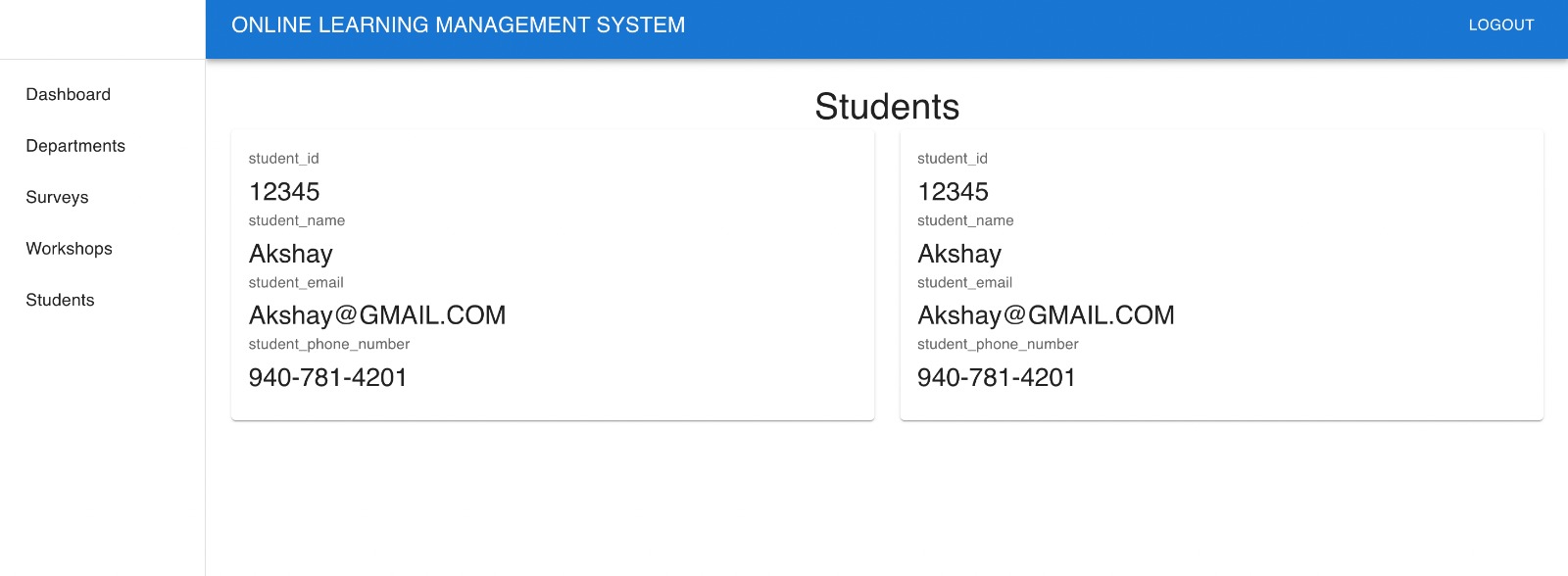


Figure 6

On clicking the students tab, the results are produced which tells who are the students working under this professor

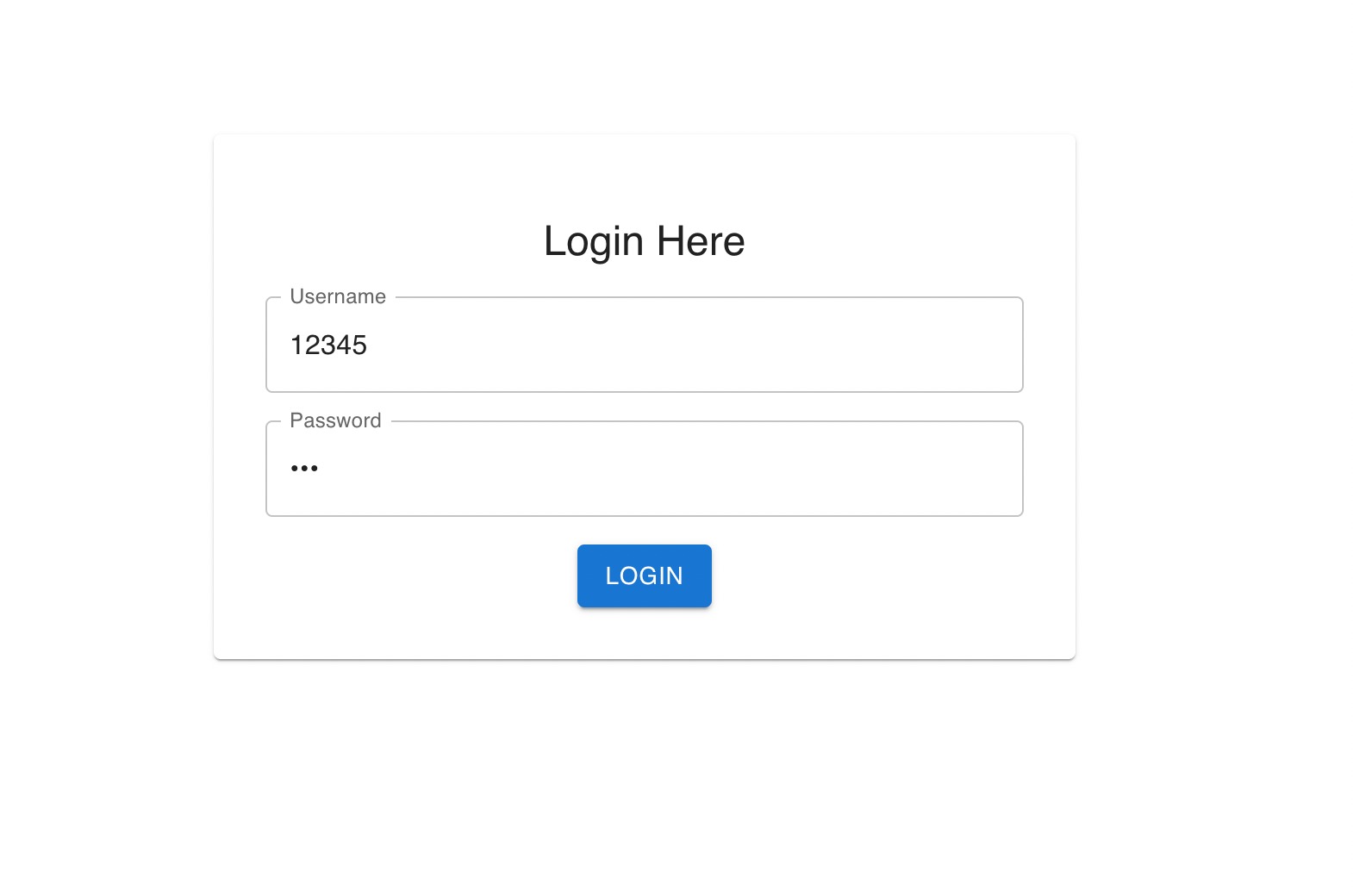


Figure 7

The above is the login page, which is common for both student or instructor. On entering the username and password in the backend it is verified whether it belongs to the student or instructor. Based on that it redirects to the next page. Now if the entered credentials are of student. The below figure 8 appears.

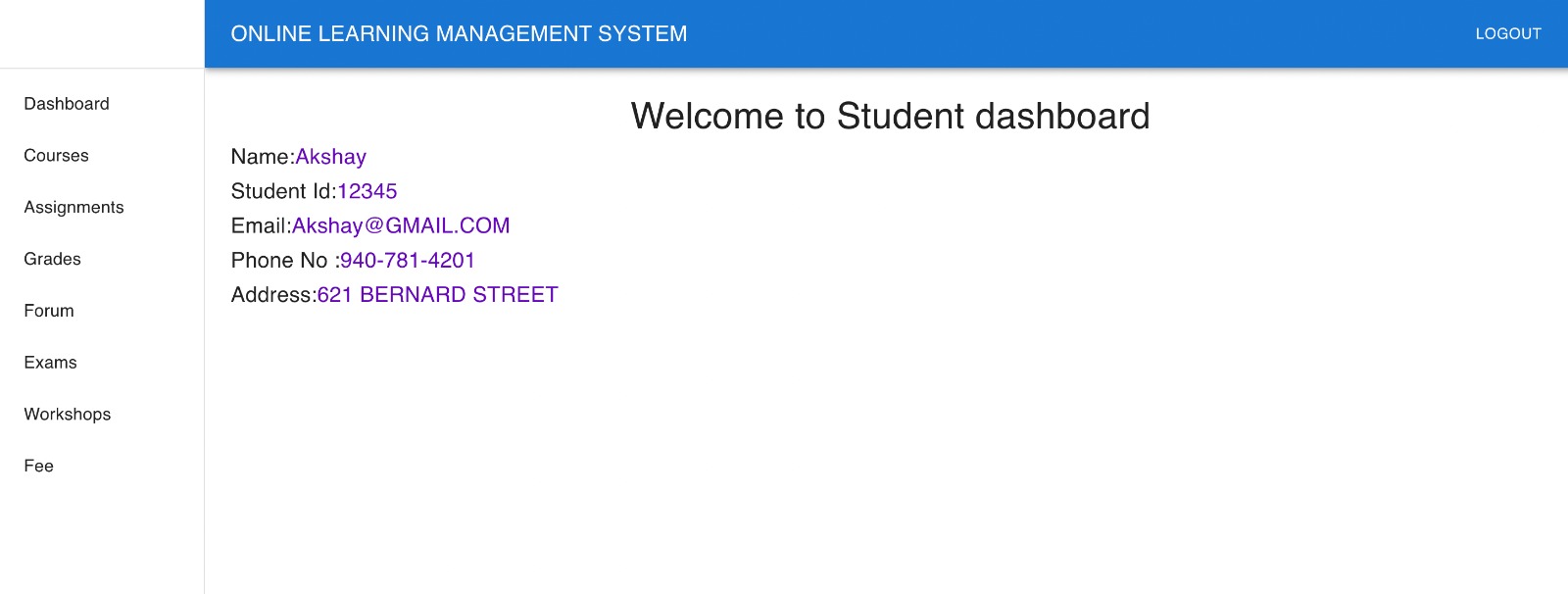


Figure 8

On entering the student details, the side pane appears which gives the various features to the students.

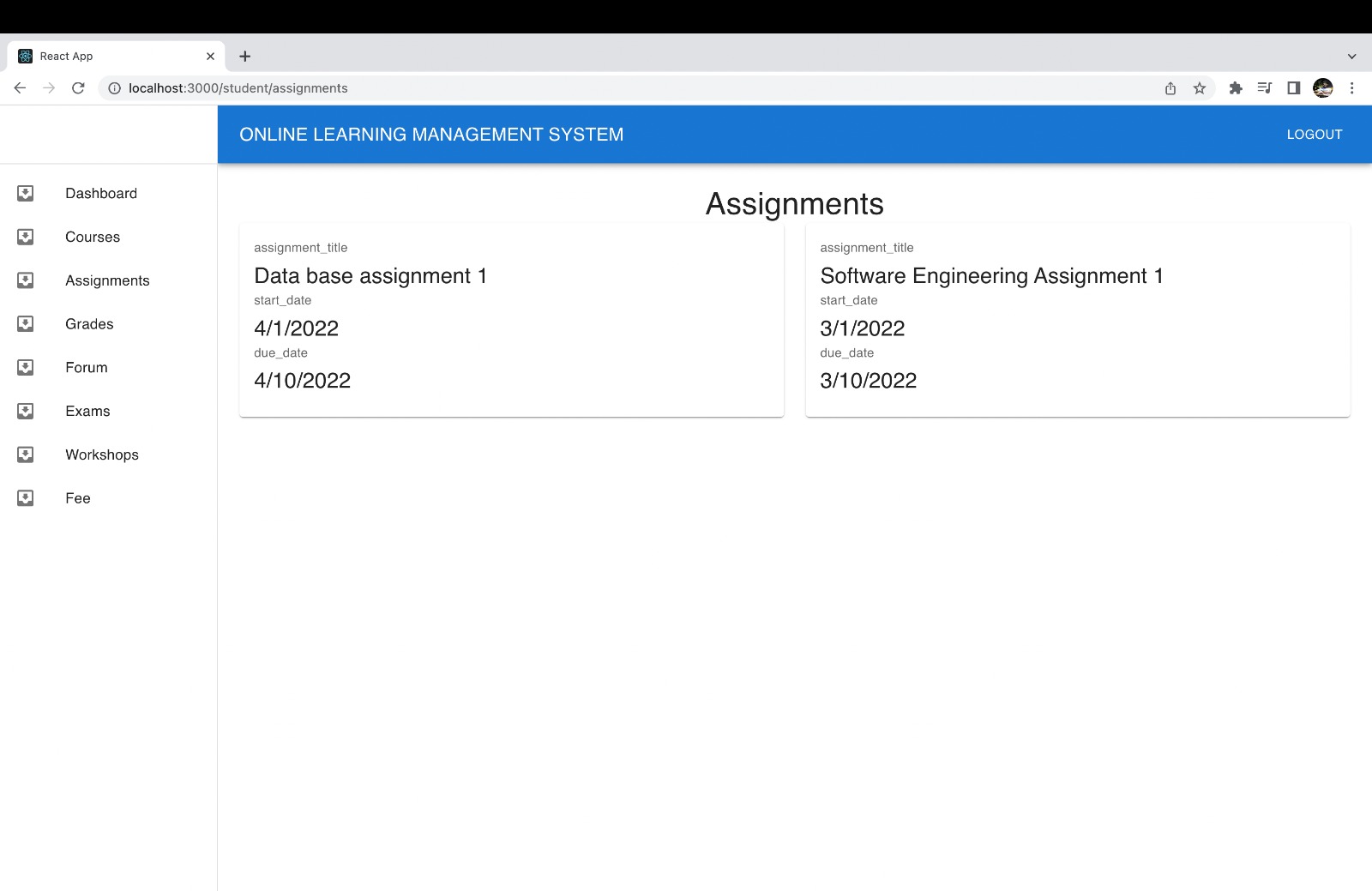


Figure 9

On clicking the assignments tab, the following details appears as Figure 9 which tells what assignments are there for student example “AKSHAY”.

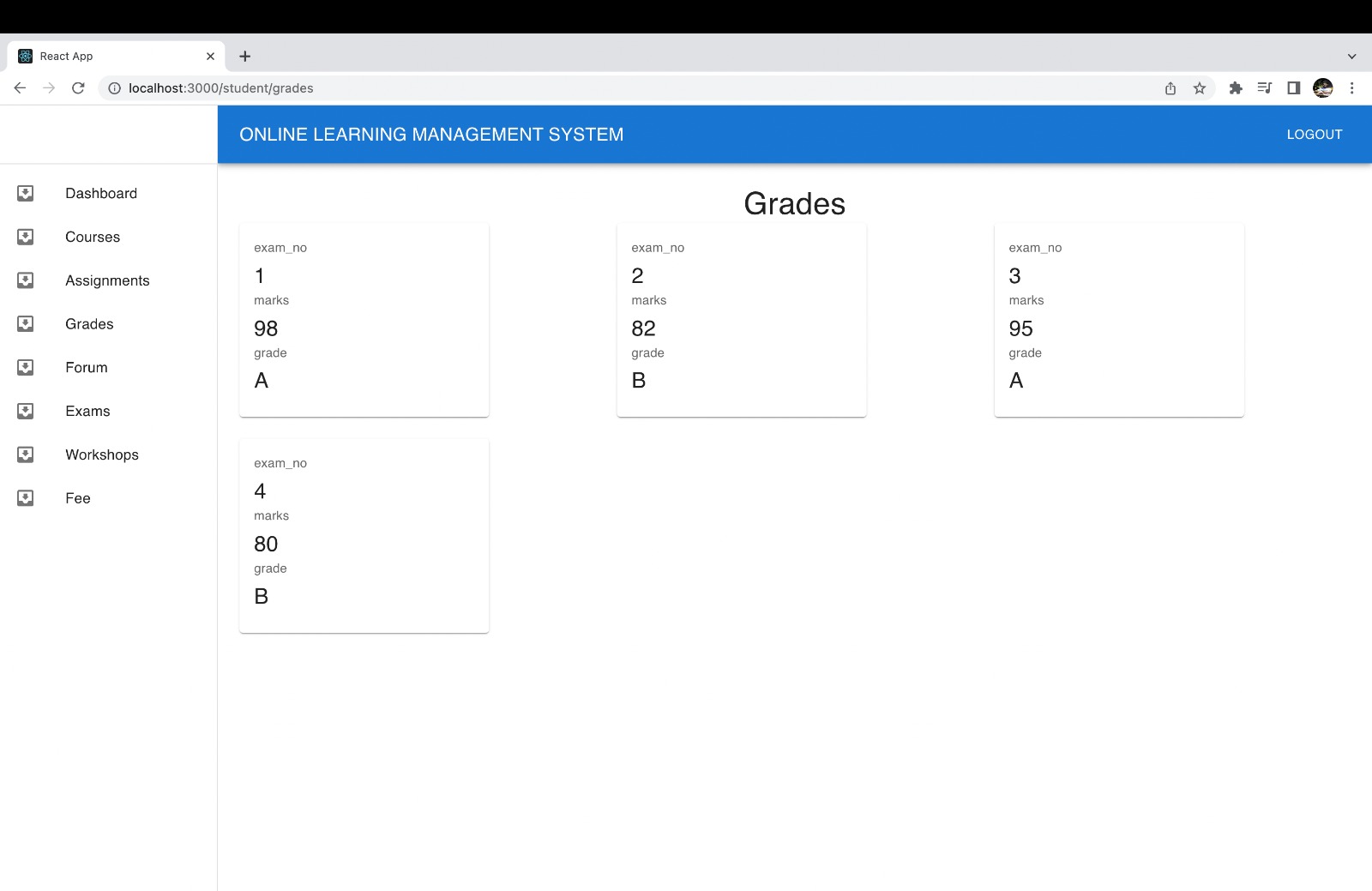


Figure 10

On clicking the grades tab, the exam number, marks and the grades appears for the all the exams written by the student.

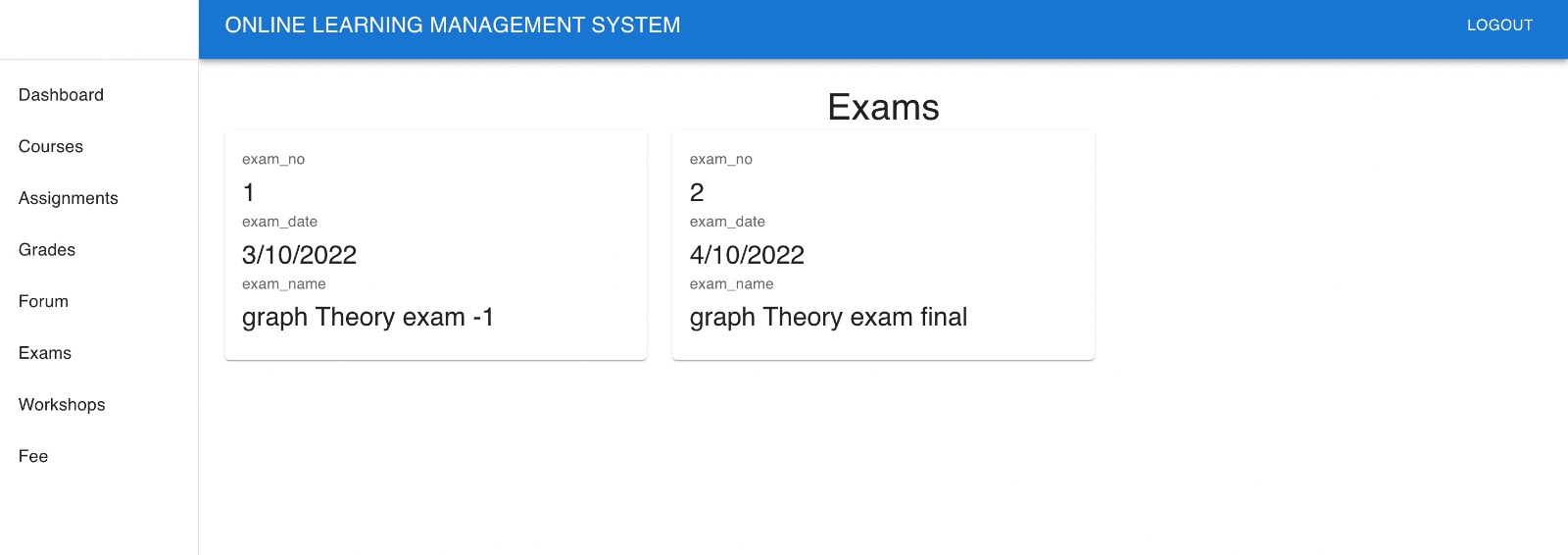


Figure 11

On clicking the exams tab, the details appear which explains what exams will be there on which date.

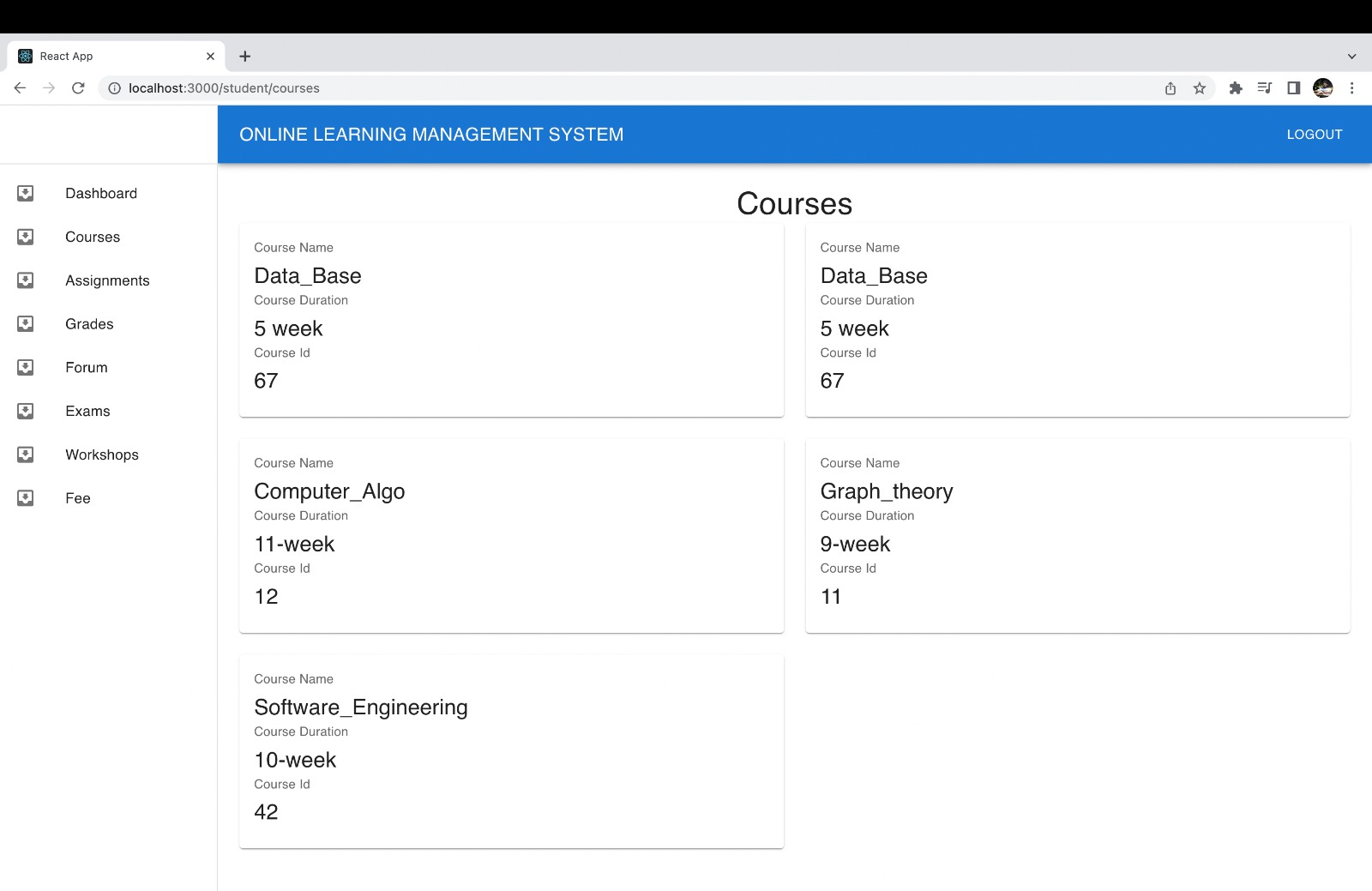


Figure 12

On clicking the courses tab, the courses details appear which tells, what courses did the student enrolled into.

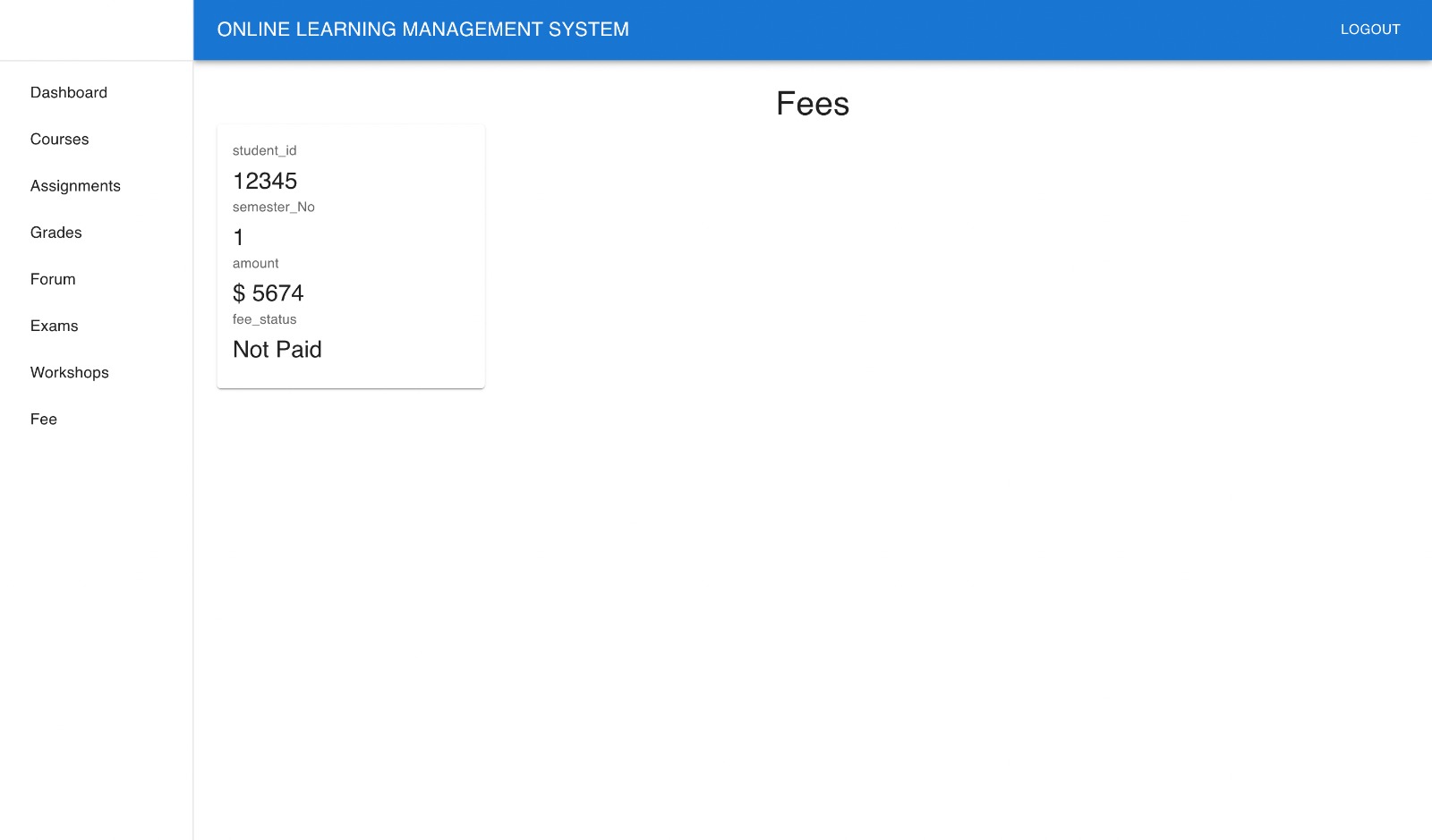


Figure 13

On clicking the fee tab, figure 13 appears which tells the amount to be paid and gives the fee status.

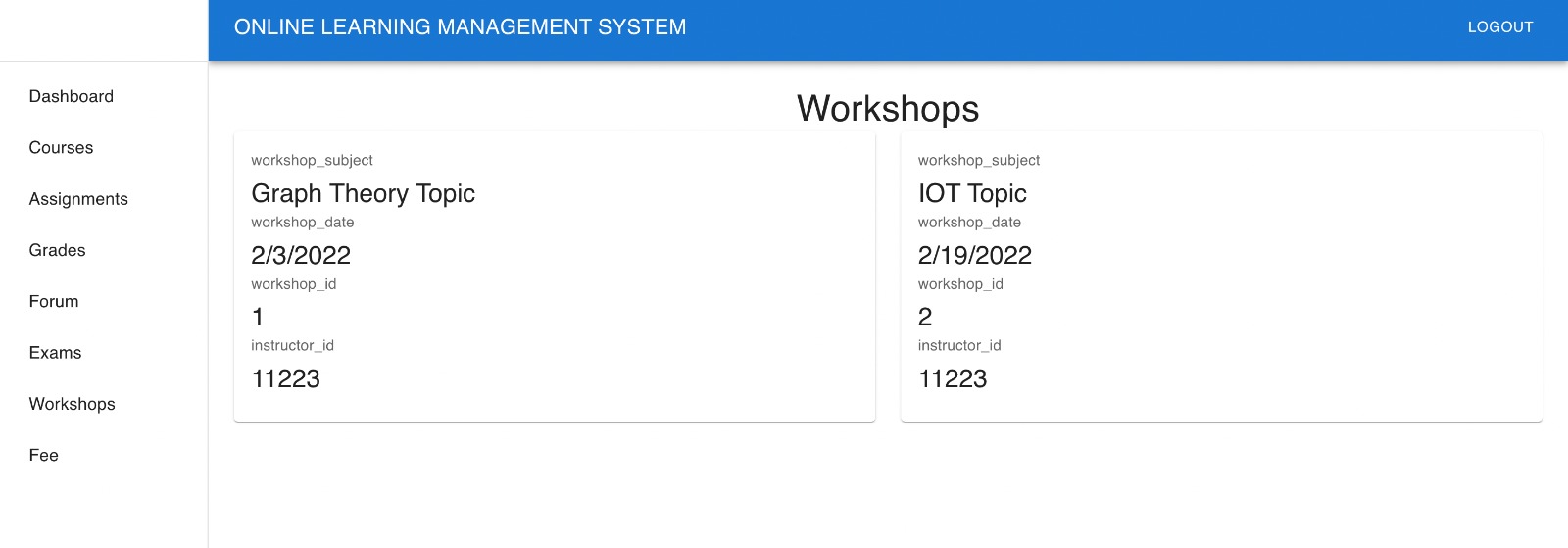


Figure 14

On clicking the workshops tab, figure 14 appears which tells different workshops for the student to attend.

1. **Execution Process –**

**How to run the Backend –**

* Need to have installed java, intellIj idea. Open pom.xml file and open it as a project. It will auto import all the required dependencies.
* In the application.properties need to change the Db schema details like schema name, user name and password.
* Right click on the Application main and run the project, then it will be running on the port 8080

**How to run the UI -**