



ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ, ಬೆಳಗಾವಿ  
VISVESVARAYA TECHNOLOGICAL UNIVERSITY - BELAGAVI

A MINI PROJECT REPORT ON

## *Online Course Management System*

*Submitted to Visvesvaraya Technological University in partial fulfillment of the requirement  
for the award of degree of*

*Bachelor of Engineering*

*in*

*Computer Science and Engineering.*

### **Submitted by:**

Mr.Manjush V G	4JN19CS050
Mr.Murali venkat J	4JN19CS057
Mr.Ganesh prasad M	4JN19CS033
Mr.Mahendra S M	4JN19CS048

### **Under the Guidance of:**

**Mr. Sathyanarayana S** BE, M.Sc(IT), M.Tech  
Assistant Prof., CS&E

**Mrs. Namitha M V** BE, M.Tech  
Assistant Prof., CS&E



Department of Computer Science & Engineering  
**Jawaharlal Nehru New College of Engineering**

Shivamogga - 577 204

March 2022

National Education Society ®



# Jawaharlal Nehru New College of Engineering

Department of Computer Science & Engineering

## CERTIFICATE

*This is to certify that the Mini-project Report entitled*

### *Online Course Management System*

#### Submitted by:

Manjush V G	4JN19CS050
Murali venkat J	4JN19CS057
Ganesh prasad M	4JN19CS033
Mahendra S M	4JN19CS048

*Students of 5<sup>th</sup> semester B.E under the supervision and guidance towards the partial fulfillment of the requirement for award of degree of the Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2021 – 2022.*

#### Signature of Guides and HOD:

\_\_\_\_\_  
**Mr. Sathyanarayana S** BE, M.Sc(IT), M.Tech  
Assistant Prof., CS&E

\_\_\_\_\_  
**Mrs. Namitha M V** BE, M.Tech  
Assistant Prof., CS&E

\_\_\_\_\_  
**Dr. Poornima K M** BE, M.Tech, Ph.D.  
Prof. & Head, Dept. of CS&E

Name of Examiners

Signature with Date

- 1.
- 2.

## **ABSTRACT**

A database management system (DBMS) is a software package with computer programs that control the creation, maintenance, and use of database. It allows organizations to conveniently develop database for various applications by database administrators (DBAs). A database is an integrated collection of data.

The online course management system mini project provides and manages various activities of an institution such as uploading course by the faculty, paying the fee for the course by the student to the institute, and approving the payment by the institute. The online course management system software is a Django, html-CSS, JavaScript, and MySQL built system that manages the activities among student, faculty, and institute. It provides respective functionalities for students, faculties, and institutes. It allows new students, faculties, or institutes to register. And allows existing students, faculties, or institutes to login. A student can enroll for a course by paying the fees to the institute. A student can add a course to Wishlist. A faculty can upload the course to the website. An institute can approve the payment done to their account number. This project is built keeping in mind various activities related to online course management and the software automates all the required functionalities

## ACKNOWLEDGEMENT

On presenting the Database Management Systems Mini Project report on "*Online Course Management System*", I feel great to express my humble feelings of thanks to all those who have helped me directly or indirectly in the successful completion of the project work.

I would like to thank our respected guides **Mr.Sathyanarayan S**, Assistant Professor, Dept. of CS & E, and **Mrs Namitha Venkateshaiah**, Assistant Professor, Dept. of CS &E, who has helped us a lot in completing this task, for their continuous encouragement and guidance throughout the project work.

I would like to thank **Dr. Nirmala Shivanand**, Professor and Head, Dept of CSE, JNNCE, Shivamogga and **Dr. Shashidhar K Kudari**, The Principal, JNNCE, Shivamogga for all their support and encouragement.

I am grateful to Department of Computer Science and Engineering and Our institution Jawaharlal Nehru National College of Engineering and for imparting me the knowledge with which I can do my best.

Finally, I also would like to thank the whole teaching and nonteaching staff of Computer Science and Engineering Dept.

**Thank You All,**

### **Name And USN**

Manjush V G	4JN19CS050
Murali venkat J	4JN19CS057
Ganesh prasad M	4JN19CS033
Mahendra S M	4JN19CS048

# TABLE OF CONTENT

<b>1. Chapter-1: Introduction</b>	<b>1-3</b>
1.1. Overview of Online Courses	1
1.2. Applications	1
1.3. Problem Statement	1
1.4. Objectives	2
1.5. Overview of Django	2
1.6. Overview of MySQL	3
1.7. Organization of report	3
<b>2. Chapter-2: Design and Implementation</b>	<b>4-18</b>
2.1. Functional Requirements	4
2.2. Design	7
2.3. HTML	8
2.4. Django MVT System	9
2.5. Django Admin Interfaces	10
2.6. Implementation	12
2.7. APIs Used	15
2.8. Summary	17
<b>3. Chapter-3: Results</b>	<b>18-35</b>
3.1. Explore	18
3.2. Search	20
3.3. Course Details	22
3.4. Login and Register	23
3.5. Student	24-30
3.5.1. My courses	26
3.5.2. Rating	28
3.5.3. Wishlist	29
3.5.4. Payment	30
3.6. Faculty	31-32
3.6.1. My courses	31
3.6.2. Upload	32
3.7. Institute	33-34
3.7.1. Approvals	33
3.8. Form Errors	35
<b>4. Chapter-4: Conclusions and Future Scope</b>	<b>36</b>

# CHAPTER-1

## INTRODUCTION

Online course management system is the future of educational system which uses technology to bring education to the fingertips of users. Main aim of Online course management system is to maintain systematic database which stores all useful information about students, lecturers and Organization. As we move forward with technology and the World Wide Web, teachers are utilizing the internet as a means to communicate with their students. With a growing presence of online educational tools, or courseware, educators are completing the routine aspects of their jobs more efficiently with less aggravation (Moodle Message Boards, 150). While many versions of courseware exist, few programs provide teachers with all of the functionality that they need. Instructors who wish to use courseware have problems deciding between an out of the box solution versus a customized system. Additionally, computer literacy among educators does not typically include programming experience. For these reasons, a piece of software that is both customizable and easy to install is required to fit their needs. Today, the software is still cumbersome and difficult to configure. For educators all over the world, providing the best of flexibility and ease of use has been a consistent problem since the advent of online course management systems.

### 1.1. Overview Of Online Courses

**Online course** is education that takes place over the Internet. They are generally conducted through a learning management system as our project, in which students can view their course syllabus and academic progress. **Online courses** are generally self-paced, allowing for greater flexibility in completing coursework.

### 1.2. Applications

As everything is being digitalized the online course management application promotes the online courses and attracts the students towards these courses. This grows the skills and knowledge of the students. It helps the students to understand the subject in depth and crack exams. This helps faculties to reach a large number of students.

### 1.3. Problem Statement

Nowadays' the world of education and learning is moving towards online. Every student may not have a good bandwidth and they may face network issues if the teaching is live. So, for those students who do not have a good bandwidth, concepts will not be clear.

## 1.4. Objectives

- Main objective of Online Course management system is to provide simple, universal access to information and services for students, faculty, and universities.
- Our aim is to give best service to the users who wants to get knowledge from our online platform.
- Enhance the quality of learning and teaching.
- Meet the learning style or needs of students.
- Improve the efficiency and effectiveness.
- Improve user-accessibility and time flexibility to engage learners in the learning process.

## 1.5. Overview of Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application. MVT Structure has the following three parts – Model: The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres). View: The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/JavaScript and Jinja files. Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

### Features of Django: -

- **Rapid Development:** - Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.
- **Secure:** - Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.
- **Scalable:** - Django is scalable in nature and has ability to switch from small quickly and flexibly to large scale application project
- **Fully loaded:** - Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.
- **Versatile:** - Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build

various types of applications like content management systems, social networks sites or scientific computing platforms etc.

- **Open Source:** - Django is an open-source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.
- **Vast and Supported Community:** - Django is a one of the most popular web frameworks. It has widely supportive community and channels to share and connect.

## 1.6. Overview of MySQL

SQL (Structured Query Language) queries. Architecture of MYSQL describes the relation among the different components of MYSQL System. MYSQL follow Client-Server Architecture. It is designed so that end user that is Clients can access the resources from Computer that is server using various networking services. The Architecture of MYSQL contain following major layer's:

- Client
- Server
- Storage Layer

### Features of MySQL: -

- MySQL is a relational database management system. This database language is based on the SQL queries to access and manage the records of the table.
- MySQL is easy to use. We must get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.
- MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.
- MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.
- MySQL is considered one of the very fast database languages, backend by many the benchmark test.

## 1.7 Organization of the report

This section deals with the Introduction and organization of the project report. Chapter 2 discusses the Specific to the Problem-Requirement Analysis-Design-Implementation which mainly consisting of ER diagrams and Schema diagram. Chapter 3 discusses the Results which consists of screenshots of database and pages of our project. Chapter 4 will deals with Conclusions and Future Scope which have information about The problem in live class is that there is no guaranteed bandwidth which causes packet loss and this effects the continuity of the class.



## **CHAPTER-2**

### **DESIGN AND IMPLEMENTATION**

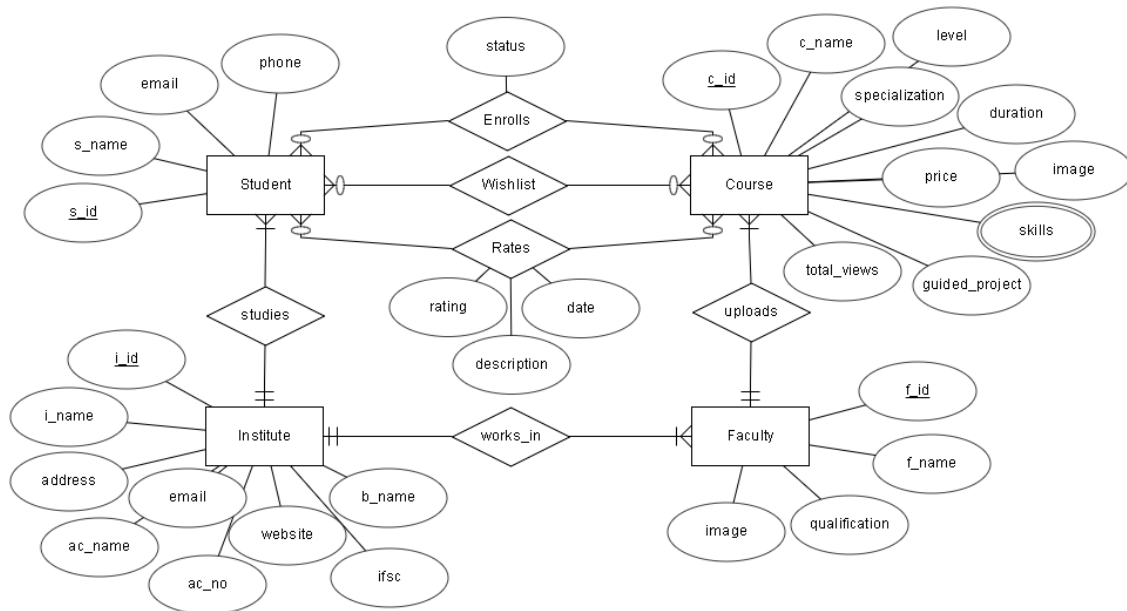
In order to have ensured a proper implementation that fulfills the needs for an OfCourse! installation package, appropriate preliminary planning took place. Initially, the project team carefully studied the source code of the system. It was important to have a full understanding of how it worked, particularly as it relates to the setup routines. Once the program was understood, a coherent plan of action was assembled.

#### **2.1. Functional Requirements**

- **Search**  
Search feature will allow a user to find the courses based on the topic, specialization, skills that he/she wants to acquire, instructor who taught the course or the courses which belong to an institute.
- **Wishlist**  
A student can add courses to Wishlist. Later he/she can enroll that course. A Wishlist allows user to create personalized collections of videos they want to buy and save them in their user account for future reference. Wishlist signifies a customer's interest in a subject without an immediate intent to purchase.
- **Enroll**  
A student could enroll to any course, if the course provided by the same institute to which the student belongs then the course will be enrolled for free else, they must pay the fees for the course. Course enrollment refers to the process of adding, changing grading options or credit hours, or dropping classes.
- **Progress**  
This website will also keep track of students' progress in the course. If the student watches up to 95% of the video, then that video will be marked as completed.
- **Upload**  
A faculty can upload courses to the website. After uploading he/she can see the number of views for his/her course, number of students who have enrolled for the course and the rating of the course. Uploading of video is done by lecturers of particular subject they belong to and students can view the uploaded videos of their interest.
- **Approve**  
An Institute must verify whether the student studies in their institute or not and the institute has to approve the enrollment of the course for a student after the payment.

## ER Diagram

The below figure shows the ER diagram and ERD is a model that identifies the concepts or entities that exist in a system and the relationships between those entities.



**Figure 2.1**

## Entities

- Institute: -**  
 This entity stores Institute details such as name, address, email, image, institute website link, account number, IFSC, account holder name. *i\_id* is the primary key. email is used for login. Bank account details is needed because the students must pay the fees for the institute to get access for a course.
- Faculty: -**  
 This entity stores the faculty details such as name, qualification, image, *i\_id*. Here *f\_id* is the primary key. *i\_id* is the foreign key which refers to the institute to which the faculty works for. Faculty will upload the courses.
- Student: -**  
 This entity stores Student details such as name, email, phone number. *s\_id* is the primary key. *i\_id* is the foreign key which refers to the institute in which the student studies. A student can add a course to Wishlist, and he/she can enroll for the course by paying the fees. After completing the course, he/she can give the rating for that course which would help other students.

- **Course: -**

This entity stores many attributes related to a course such as name, description, playlist\_id, specialization, level, skills, duration, price, image, total\_views, guided\_project, date, no\_videos, rating. c\_id is the primary key. f\_id is the foreign key which refers to the faculty who has uploaded the course. playlist\_id stores the playlist id of the YouTube playlist which is unlisted (playlist can be accessed only through link). skills attribute is multivalued attribute. no\_videos will store the number of videos in the playlist, duration will store the total duration of the playlist and values of both attributes are calculated before inserting a course using the YouTube data API v3. guided\_project is Boolean attribute which will store 1 if the course is guided project else 0. Initially rating will be zero, when a student gives a review then the trigger which we have created will be fired and the rating will be updated.

### Relationship between Entities

- Student—Enrolls—Course

	<b>Constraint</b>	<b>Cardinality ratio</b>
<b>Student</b>	Optional	Many
<b>Course</b>	Optional	Many

- Student—Wishes—Course

	<b>Constraint</b>	<b>Cardinality ratio</b>
<b>Student</b>	Optional	Many
<b>Course</b>	Optional	Many

- Student—Rates—Course

	<b>Constraint</b>	<b>Cardinality ratio</b>
<b>Student</b>	Optional	Many
<b>Course</b>	Optional	Many

- Student—Studies in—Institute

	<b>Constraint</b>	<b>Cardinality ratio</b>
<b>Student</b>	Mandatory	Many
<b>Institute</b>	Mandatory	One

- Faculty—Works in—Institute

	<b>Constraint</b>	<b>Cardinality ratio</b>
<b>Faculty</b>	Mandatory	Many
<b>Institute</b>	Mandatory	One

- Faculty—Uploads—Course

	Constraint	Cardinality ratio
Faculty	Mandatory	One
Course	Mandatory	Many

## 2.2. Design

### Schema diagram:

- Schema diagram is a skeletal structure that represents the local view of the entire database.
- Pointed arrow is used here to represent the foreign keys of tables (attribute at the head of the arrow represents parent key and attributes at the tail represents child key).

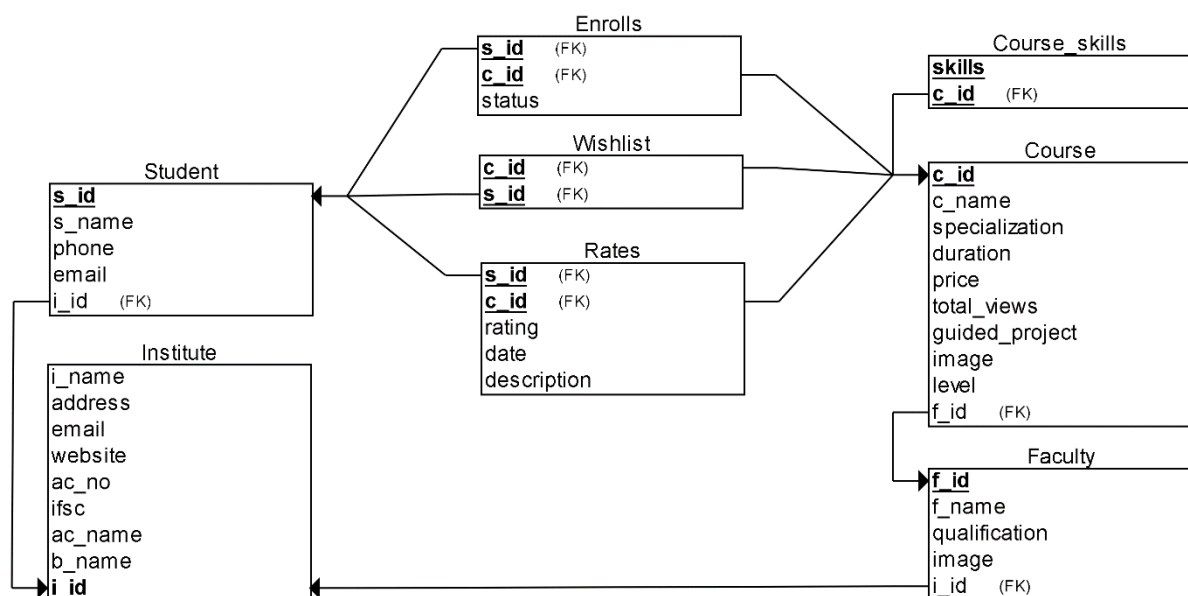


Figure 2.2

## 2.3. HTML

- HTML stands for Hyper Text Markup Language.
- HTML is the standard markup language for creating Web pages.
- HTML describes the structure of a Web page.
- HTML consists of a series of elements.
- HTML elements tell the browser how to display the content.

### HTML TAGS

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document.
- The `<html>` element is the root element of an HTML page.
- The `<head>` element contains meta information about the HTML page.
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab).
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.,
- The `<h1>` element defines a large heading.
- `<p>` element defines a paragraph.

#### Examples: -

`<p>Paragraph</p>`

`<h2>Heading tag</h2>`

`<b>Bold</b>`

`<i>Italic</i>`

`<u>Underline</u>`

`<br>`Tag: br stands for break line, it breaks the line of the code.

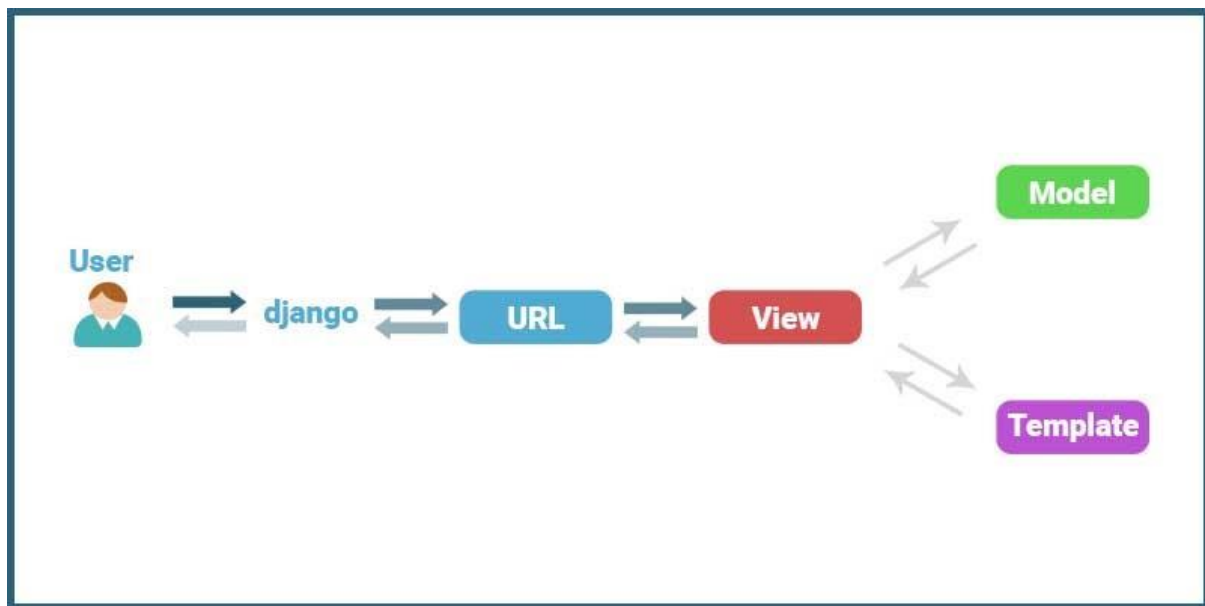
`<hr>` Tag: hr stands for Horizontal Rule. This tag is used to put a line across the webpage

### W3.CSS

- W3.CSS is a modern framework with built-in responsiveness and easy to learn and use compared to other CSS frameworks.
- It aims to speed up and simplify web development and support modern responsive devices like Mobile, Laptop, Tablet and Desktop.
- W3.CSS was designed to be a high-quality alternative to Bootstrap.

- W3.CSS is free for everyone to use. Anyone can use it by simply linking the style in their webpage.

## 2.4. Django MVT System



**Figure 2.3**

The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model, View and Template.

**Model:** This is an abstraction layer for structuring and manipulating the data of the Web Application. It acts as an interface for maintaining data. This is a logical data structure behind the entire application and helps to handle the database.

**View:** This layer encapsulates the logic responsible for processing a user's request and returns a response. It is a user interface to execute the logic and interact with the models. It is responsible for displaying all or a portion of data to the user.

**Template:** The template layer provides a designer-friendly syntax for rendering the information to be presented to the user. It contains the static parts of the desired HTML output along with some special syntax, also known as Django Template Language (DTL), describing how dynamic content will be inserted.

## 2.5. Django Admin Interface

Django provides an admin site to allow CRUD (Create Read Update Delete) operations on registered app model. It is a built-in feature of Django that automatically generates interface for models. We can see the URL entry for admin in urls.py file, it is implicit and generated while creating a new project.

1. urlpatterns = [
2. path('admin/', admin.site.urls),
3. ]

The admin app(django.contrib.admin) is enabled by default and already added into the INSTALLED\_APPS list present in the settings.py file.

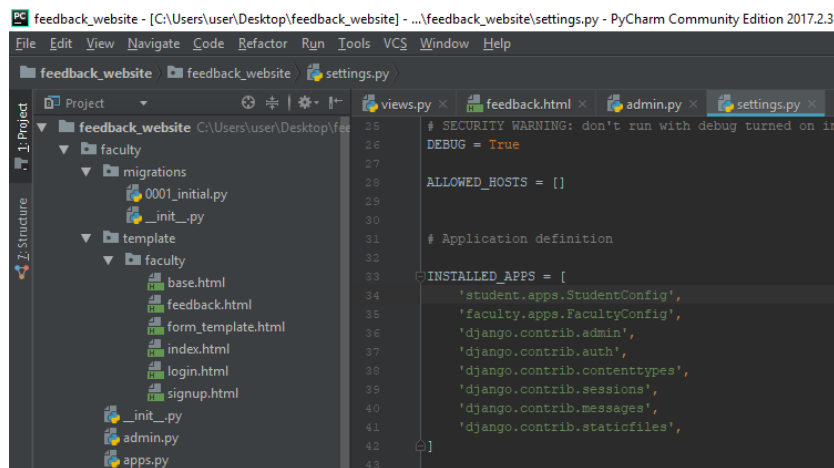


Figure 2.4

It can be easily accessed by after login from the admin panel, lets run the server python3 manage.py runserver and access it through the localhost:8000/admin.

A login form will be displayed,

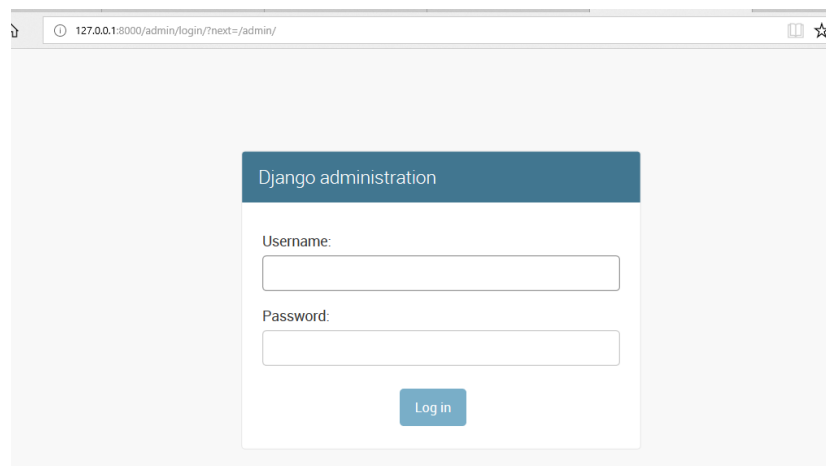
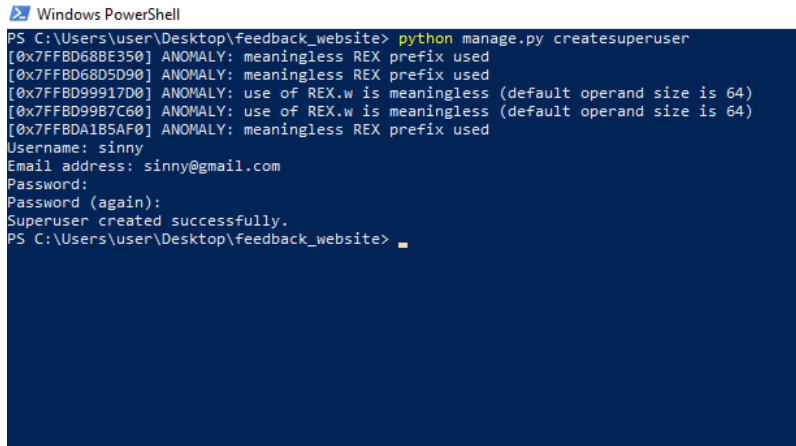


Figure 2.5

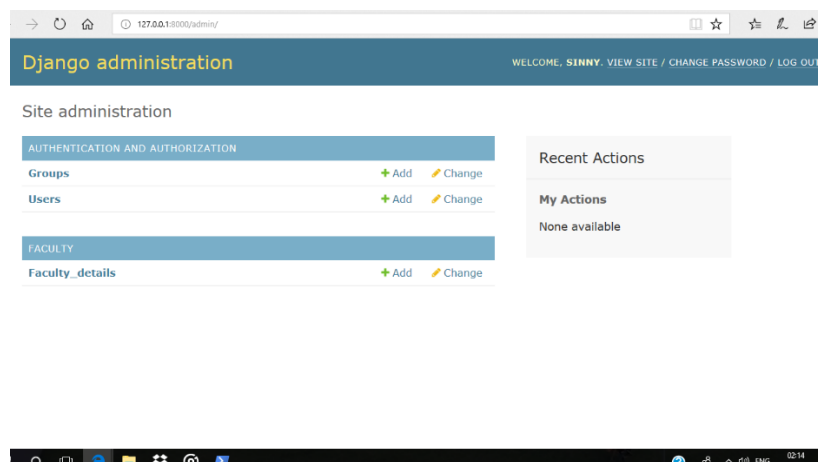
To login, first create admin (super user) user and provide password as we did here:



```
Windows PowerShell
PS C:\Users\user\Desktop\feedback_website> python manage.py createsuperuser
[0x7FFBD688E350] ANOMALY: meaningless REX prefix used
[0x7FFBD68D5D90] ANOMALY: meaningless REX prefix used
[0x7FFBD99917D0] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBD9987C60] ANOMALY: use of REX.w is meaningless (default operand size is 64)
[0x7FFBDA185AF0] ANOMALY: meaningless REX prefix used
Username: sinny
Email address: sinny@gmail.com
Password:
Password (again):
Superuser created successfully.
PS C:\Users\user\Desktop\feedback_website>
```

**Figure 2.6**

Super user is created successfully, now you can login with the username and password that you have given while creating the superuser. After login you will see a page as shown below:



**Figure 2.7**

It is an admin dashboard that provides facilities like creating groups and users. It also used to manage the models.

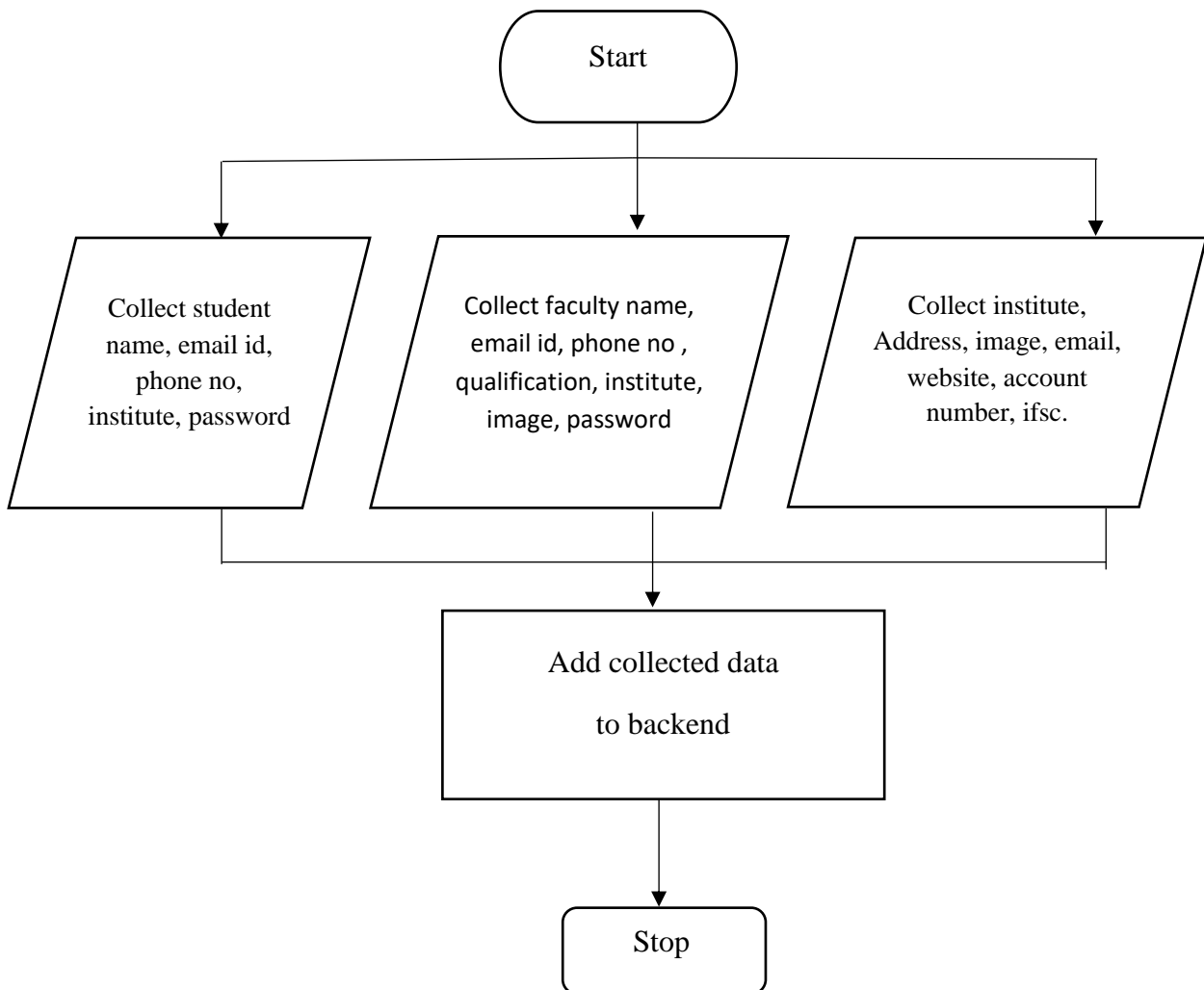
This interface will let you administrate Django groups and users, and all registered models in your app. The interface gives you the ability to do at least the "CRUD" (Create, Read, Update, Delete) operations on your models.

One of the most powerful parts of Django is the automatic admin interface. It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site. The admin's recommended use is limited to an organization's internal management tool. It's not intended for building your entire front end around.



## 2.6. Implementation

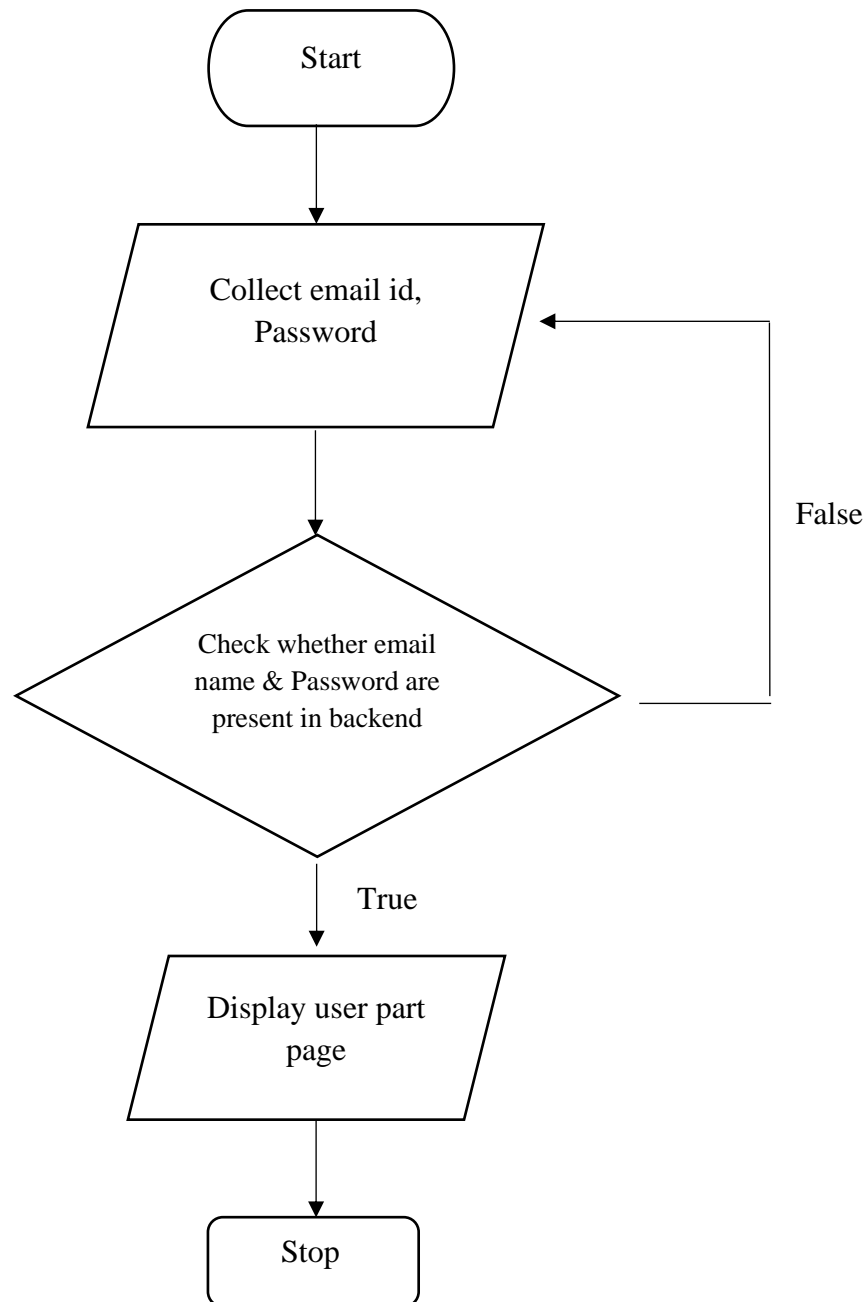
### 2.6.1 Flowchart for Student, Faculty, Institute registration:



**Figure 2.8**

This flow chart shows the registration of the student, faculty, and institute. They have to give related details such as name, email id. In addition, student have to give their phone number and select the institute in which they study. The faculty has to give their image, qualification and the institute in which they work. The institute has to provide payment details such as account number and ifsc. If any detail is wrong, then the user will be given error message and asked to give the correct details once again.

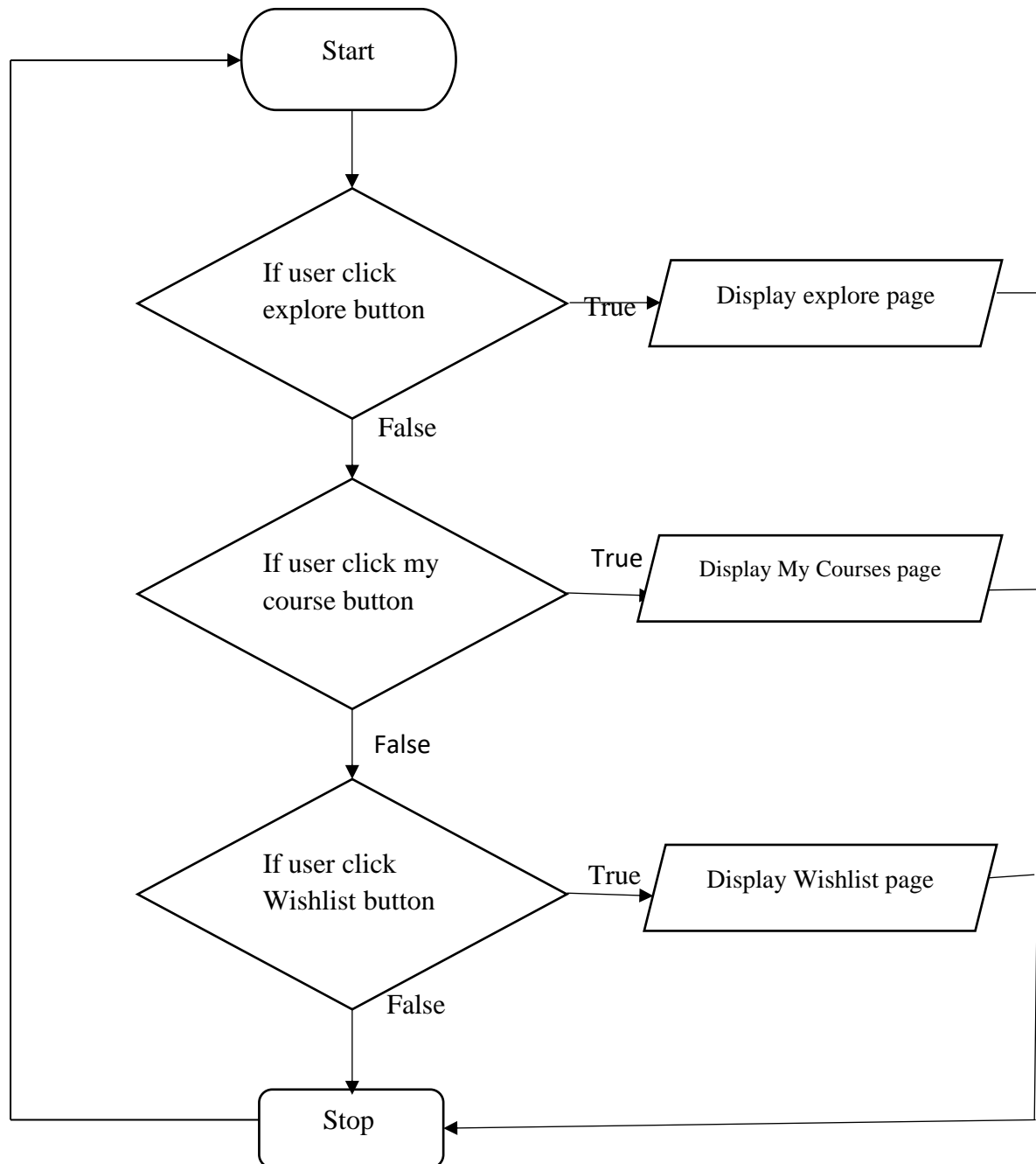
### 2.6.2 Flowchart for user login



**Figure 2.9**

This flow chart shows the login of student, faculty, and institute. The user has to give their email id and password which they have given during the registration. Then the server will verify the details given, if the password didn't match or if the given email id is not registered then it will give the error message. If the given email id is registered and the password matches, then the user will be logged in and he will be redirected to explore page.

### 2.6.4 Flowchart for user part page:



**Figure 2.10**

This flow chart shows the navigation of the user to different pages. When a student logs in to this website then the navigation bar will get two more extra tabs which are My Courses, and Watchlist. If the student clicks on My Courses, then he will be redirected to a page which has a list of courses that he has enrolled. Similarly, if he clicks on Wishlist then he will be redirected to a page which has a wished list of courses.

## 2.7. APIs used

YouTube Data API V3	
APIs used	Explanation
playlistItems().list()	<p>In this function we have passed four parameters: part, playlistId, maxResults and pageToken.</p> <p>part is string which contains comma separated playlistItem resource properties. We have used contentDetails (to get video ids of all videos in the playlist) and snippet (to get title, description, and thumbnail of each video of the playlist) properties in part.</p> <p>With a single request we can fetch maximum of 50 objects of the given properties. If the playlist has more than 50 videos, then we will get nextPageToken (page token of next page) in the response. Now we can use this token as a parameter to fetch next 50 videos.</p>
Videos().list()	<p>In this function we have passed two parameters: part, id.</p> <p>Here part contains videos resource properties. We have used only one property that is contentDetails (to get duration of video whose id is given).</p> <p>id is string which contains comma separated video ids. We can fetch duration of multiple videos (maximum of 50) at a single time.</p>

**Table 2.1**

YouTube Player API	
APIs used	Explanation
YT.Player()	<p>This function is used to create a Player object. This function will take two parameters: first parameter is DOM element or id of HTML element where the API will insert &lt;iframe&gt; tag containing the player, and second parameter is a JavaScript object which specifies the player options.</p> <p>The object contains the properties such as width, height, videoId, playerVars, events.</p>

	<p>playerVars is an object which contains player parameters such as rel (related videos), autoplay, controls etc. These parameters are either set to 0 or 1.</p> <p>Events is an object which contains events and their corresponding handler functions.</p>
playerObj.getDuration()	Here playerObj is the object that we created using above function. The function getDuration will return the duration of the playerObj video in seconds.
playerObj.getCurrentTime()	This function will return the current playing time of the playerObj video in seconds.

**Table 2.2**

<b>Django APIs</b>	
<b>APIs used</b>	<b>Explanation</b>
request.GET.get	request.GET contains the GET variables. These are what you see in your browser's address bar. The. get() method is a method used for dictionaries .
Request.POST.get	request.POST contains the POST variables. This function is used to get the variables that are sent with http post request.
objects.filter	Objects.filter is to get a matching result from the database, return a list of objects.
objects.get	It is a method of accessing objects in python. It returns the value associated with that object.
objects.raw	<p>This method takes a raw SQL query, executes it, and returns a django.db.models.query.RawQuerySet instance.</p> <p>This RawQuerySet instance can be iterated over like a normal QuerySet to provide object instances.</p>
objects.set_cookie	The set_cookie() method in DjangoHttpResponse has a name: Name of the cookie.value: Value you want to store , int or string but it will return string.
request.COOKIES.get	Using request.COOKIES.get () Django also provides a method to get the desired value from the cookie. You can directly access that value using get method over request object.

objects.latest	Returns the latest object in the table based on the given field(s).
objects.create	To create and save an object in a single step, we can use the create() method.

**Table 2.3**

## 2.8. SUMMARY

An online course management system (OCMS) is a collection of software tools providing an online environment for course interactions. An OCMS typically includes a variety of online tools and environments, such as: An area for faculty posting of class materials such as course syllabus and handouts. Main aim of OCMS is to provide area for students to access videos of subjects they want and to provide area for lecturers to upload their subject videos. In this chapter we have discussed about the design and implementation of the project. It involves the system design, ER diagram, schema diagram and system implementation. System design involves developing several modules of system at different levels of abstraction. ER diagram displays the database in the form of entities and relationships. ER diagrams consisting of all entities and its relationship is mentioned in this chapter with all flowcharts and it's description. Schema diagram contains tables with their attributes that defines the schema and flowchart of schema is all mentioned with all keys present. This chapter also contains information about APIs used in developing the project. YouTube Data API V3, Django APIs are the main APIs which are used in project. The program also wrote the database connection information to various files in both the Django language. In order to properly access the database, every file that referenced this connection information had to be modified at install time. Lastly we have discussed about the implementation with all the respective pages.

# CHAPTER 3

## RESULTS

### 3.1 Explore

The screenshot shows the 'Explore' page of an Online Course Management System. At the top, there is a search bar and a 'login' button. Below the search bar, there are five category buttons: 'Web development', 'Java', 'Computer Science', 'Internet of Things', and 'Electronics and Communication'. The main content area is divided into two sections: 'Most Viewed' and 'Recently Launched'. Each section displays a grid of course cards. Each card includes a thumbnail image, a title, a difficulty level, a duration, a price, and a rating. The 'Most Viewed' section includes courses like 'Unix Programming', 'Operating System', 'Database Management System', 'Automata and Theory of Computation', and 'Signals and Systems'. The 'Recently Launched' section includes 'Signals and Systems', 'Microcontroller and Embedded Systems', 'Automata and Theory of Computation', 'Unix Programming', and 'Operating System'.

**Figure 3.1**

It is the main page of our mini project which contains student, faculty, institute login button and complete course details.

	c_id	c_name	specializat	image	duration	price	tota	f_id	leve	guide	playlistid	description	date	rating	no_videos
<input type="checkbox"/>	3	Database Management System	Web development	course/DBMS.png	20:55:27	900	32	4	Begin	0	PLbVT2Xgpgf	A database management system (or DBMS) is essential...	2022-01-16	3	31
<input type="checkbox"/>	4	Object Oriented Concepts in Java	Java	course/OOCjava.jp	32:46:45	1200	12	8	Begin	0	PLbVT2Xgpgf	Learn fundamental features of object oriented lang...	2022-01-16	3	48
<input type="checkbox"/>	5	Operating System	Computer Science	course/Operating_	34:15:19	850	84	6	Inter	0	PLbVT2Xgpgf	Introduce concepts and terminology used in OS. Exp...	2022-01-16	0	46
<input type="checkbox"/>	6	Unix Programming	Computer Science	course/unix.jpg	28:24:1	1300	86	3	Inter	0	PLbVT2Xgpgf	Interpret the features of UNIX and basic commands....	2022-01-16	3	47
<input type="checkbox"/>	7	Automata and Theory of Computation	Computer Science	course/ATC.png	27:57:37	1600	19	6	Adva	0	PLbVT2Xgpgf	Introduce core concepts of ATC. Identify different...	2022-01-24	5	52
<input type="checkbox"/>	8	Microcontroller and Embedded Systems	Internet of Things	course/MCES.jpg	12:3:30	1100	6	9	Begin	0	PLbVT2Xgpgf	Understand the fundamentals of ARM based systems, ...	2022-01-25	0	8
<input type="checkbox"/>	9	Signals and Systems	Electronics and Communication	course/s_and_s.jp	8:21:32	850	15	10	Inter	0	PL3l2AF3mjL Su4wo1NfxW	Definition of signal and systems, communication an...	2022-02-09	0	51

Figure 3.2

This picture shows the course table. This table consists of course id and course name, specialization, image, duration of course, price, faculty id, levels whether it is for beginners, intermediate or advanced, guided projects, playlist\_id, description of course, uploaded date, ratings, total number of videos.





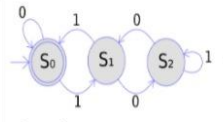




## 3.2 Search

back

Search

Courses by Jawaharlal Nehru National College of Engineering(JNNCE)


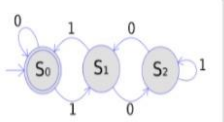
 <p>Beginners 20:55:27</p> <p><b>Database Management System</b></p> <p>A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities</p> <p>₹900 3.0 ★</p>	 <p>Beginners 32:46:45</p> <p><b>Object Oriented Concepts in Java</b></p> <p>Learn fundamental features of object oriented language in JAVA. Set up Java JDK environment to create, debug and run simple Java programs. Create</p> <p>₹1200 3.0 ★</p>	 <p>Intermediate 34:15:19</p> <p><b>Operating System</b></p> <p>Introduce concepts and terminology used in OS. Explain threading and multithreading systems. Illustrate process synchronization and concept of</p> <p>₹850 0.0 ★</p>	 <p>Intermediate 28:24:1</p> <p><b>Unix Programming</b></p> <p>Interpret the features of UNIX and basic commands. Demonstrate different UNIX files and permissions. Implement shell programs. Explain UNIX process,</p> <p>₹1300 3.0 ★</p>
 <p>Advanced 27:57:37</p> <p><b>Automata and Theory of Computation</b></p> <p>Introduce core concepts of ATC. Identify different Formal language Classes and their Relationships. Design Grammars and Recognizers for different</p> <p>₹1600 5.0 ★</p>	 <p>Beginners 12:3:30</p> <p><b>Microcontroller and Embedded Systems</b></p> <p>Understand the fundamentals of ARM based systems, basic hardware components, selection methods and attributes of an embedded system.</p> <p>₹1100 0.0 ★</p>	 <p>Intermediate 8:21:32</p> <p><b>Signals and Systems</b></p> <p>Definition of signal and systems, communication and control system as examples Classification of signals. Amplitude scaling, addition,</p> <p>₹850 0.0 ★</p>	

**Figure 3.3**

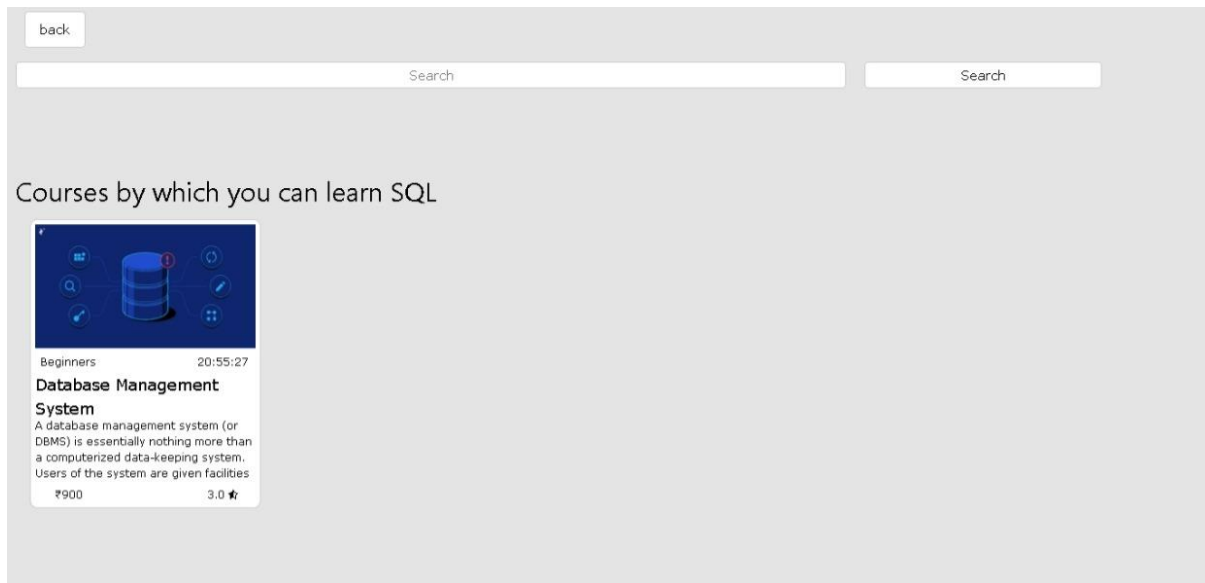
back

Search

Courses by Sankhya N Nayak

 <p>Intermediate 34:15:19</p> <p><b>Operating System</b></p> <p>Introduce concepts and terminology used in OS. Explain threading and multithreading systems. Illustrate process synchronization and concept of</p> <p>₹850 0.0 ★</p>	 <p>Advanced 27:57:37</p> <p><b>Automata and Theory of Computation</b></p> <p>Introduce core concepts of ATC. Identify different Formal language Classes and their Relationships. Design Grammars and Recognizers for different</p> <p>₹1600 5.0 ★</p>
---	---


**Figure 3.4**



**Figure 3.5**

There is a search feature which will take a keyword as an input and returns courses which has that keyword. Apart from searching the courses by their title and description, this feature will also search the courses which belong to a particular institute as shown in the **Figure 3.3**. It also searches the courses which are thought by a particular lecturer as shown in the **Figure 3.4**. It also searches the courses which teaches a particular skill as shown in the **Figure 3.5**.


### 3.3 Course Details



#### Database Management System

A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities to perform several kinds of operations on such a system for either manipulation


[Read more](#)



**Sathyanaryana S**

B.E(CSE) M.Tech (NIE)

[Read more](#)




**Jawaharlal Nehru National College of Engineering(JNNCE)**

info@jnnce.ac.in  
[website](#)

Skills you are going to be acquired :

- SQL
- Relational\_Algebra
- JDBC

 Rahul 3 ★

ok

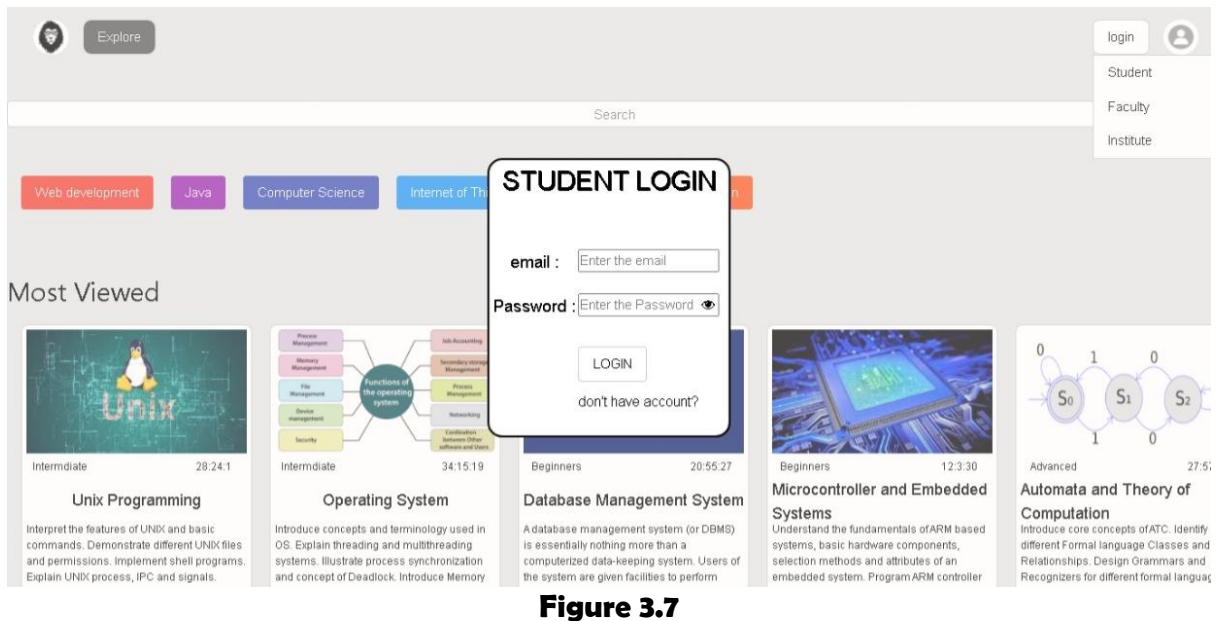
Enroll

Add to wishlist

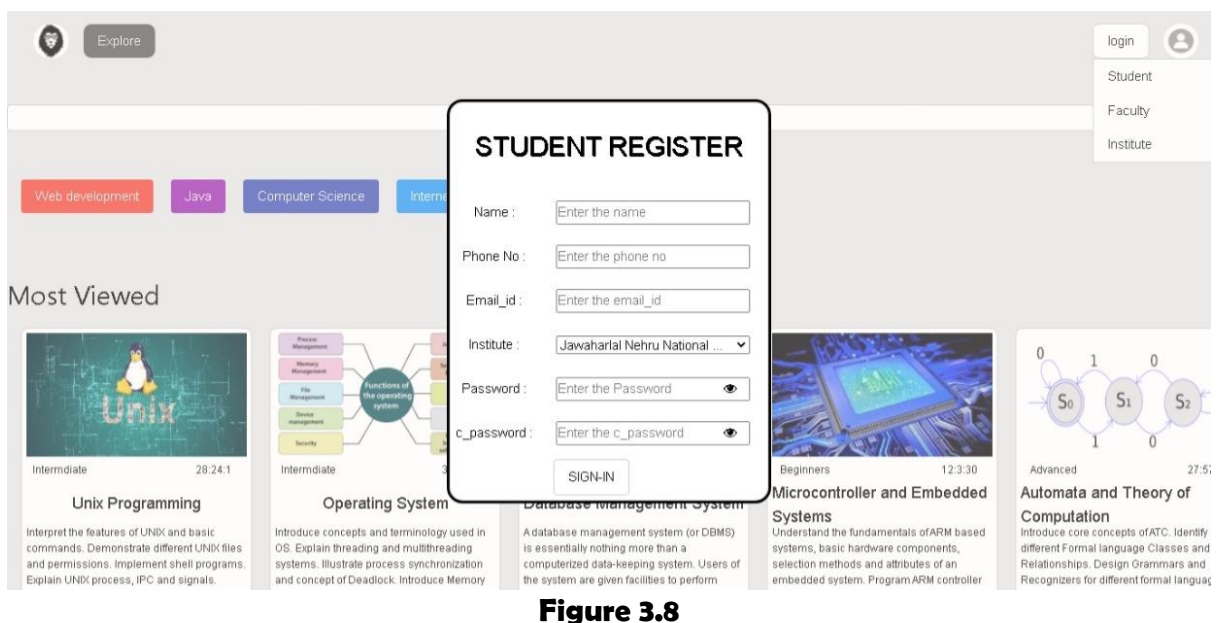
**Figure 3.6**

This page contains detailed information of a course, the faculty who has thought this course, the institute to which this course belongs, ratings and feedback details given by the students who have already enrolled for this course. Student can add this course to Wishlist or enroll to this course.

### 3.4 Login and Register



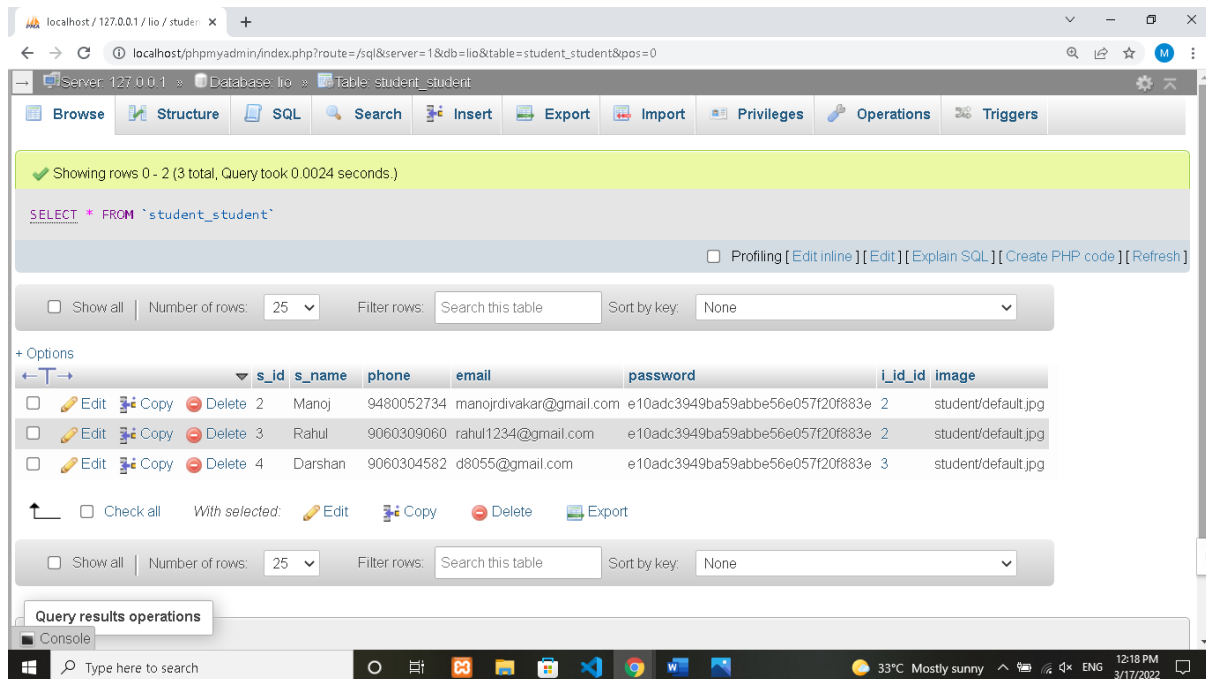
**Figure 3.7**



**Figure 3.8**

**Figure 3.7** and **Figure 3.8** shows the login and register page for students respectively. The student has login with his credentials before accessing my courses page and Wishlist page. If a student is not yet registered then he can register himself by giving some details such as name, email, phone number, and the institute in which he studies. Similarly faculty and institute can also register themselves and login.

### 3.5 Student



**Figure 3.9**

**Figure 3.9** shows the student table. This table stores students' details such as name, email, phone number, institute id of the institute in he studies, and it also stores the password which is hashed.

**Figure 3.10** shows the explore page after a student is logged in. When a student is logged in then he is redirected to explore page where his name will be displayed. This page will also give a list of courses which will be free for him. These are the courses which are from the institute in which the student studies. Every student gets the courses uploaded by his institute for free.

[Explore](#)
[My courses](#)
[Wishlist](#)
[logout](#)

[Web development](#)
[Java](#)
[Computer Science](#)
[Internet of Things](#)
[Electronics and Communication](#)

Hello Manoj

Free for Jawaharlal Nehru National College of Engineering(JNNCE)

Intermediate 28:24:1

**Unix Programming**

Interpret the features of UNIX and basic commands. Demonstrate different UNIX files and permissions. Implement shell programs. Explain UNIX process.

₹1300 3.0 ★

Beginners 20:55:27

**Database Management System**

A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities

₹900 3.0 ★

Intermediate 34:15:19

**Operating System**

Introduce concepts and terminology used in OS. Explain threading and multithreading systems. Illustrate process synchronization and concept of

₹850 0.0 ★

Advanced 27:57:37

**Automata and Theory of Computation**

Introduce core concepts of ATC. Identify different Formal language Classes and their Relationships. Design Grammars and Recognizers for different

₹1600 5.0 ★

Beginners

**Object Oriented Con in Java**

Learn fundamental features i oriented language in JAVA. S JDK environment to create, d run simple Java programs. Cr

₹1200

Most Viewed

Intermediate 28:24:1

**Unix Programming**

Interpret the features of UNIX and basic commands. Demonstrate different UNIX files and permissions. Implement shell programs. Explain UNIX process.

₹1300 3.0 ★

Intermediate 34:15:19

**Operating System**

Introduce concepts and terminology used in OS. Explain threading and multithreading systems. Illustrate process synchronization and concept of

₹850 0.0 ★

Beginners 20:55:27

**Database Management System**

A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities

₹900 3.0 ★

Advanced 27:57:37

**Automata and Theory of Computation**

Introduce core concepts of ATC. Identify different Formal language Classes and their Relationships. Design Grammars and Recognizers for different

₹1600 5.0 ★

Intermediate

**Signals and Syst**

Definition of signal and syste communication and control sy examples Classification of sig Amplitude scaling, addition,

₹850

Recently Launched

Intermediate 8:21:32

**Signals and Systems**

Definition of signal and systems, communication and control system as examples Classification of signals. Amplitude scaling, addition,

₹850 0.0 ★

Beginners 12:3:30

**Microcontroller and Embedded Systems**

Understand the fundamentals of ARM based systems, basic hardware components, selection methods and attributes of an embedded system.

₹1100 0.0 ★

Advanced 27:57:37

**Automata and Theory of Computation**

Introduce core concepts of ATC. Identify different Formal language Classes and their Relationships. Design Grammars and Recognizers for different

₹1600 5.0 ★

Intermediate 28:24:1

**Unix Programming**

Interpret the features of UNIX and basic commands. Demonstrate different UNIX files and permissions. Implement shell programs. Explain UNIX process,

₹1300 3.0 ★

Intermediate

**Operating Syst**

Introduce concepts and term used in OS. Explain threading multithreading systems. Illus process synchronization and

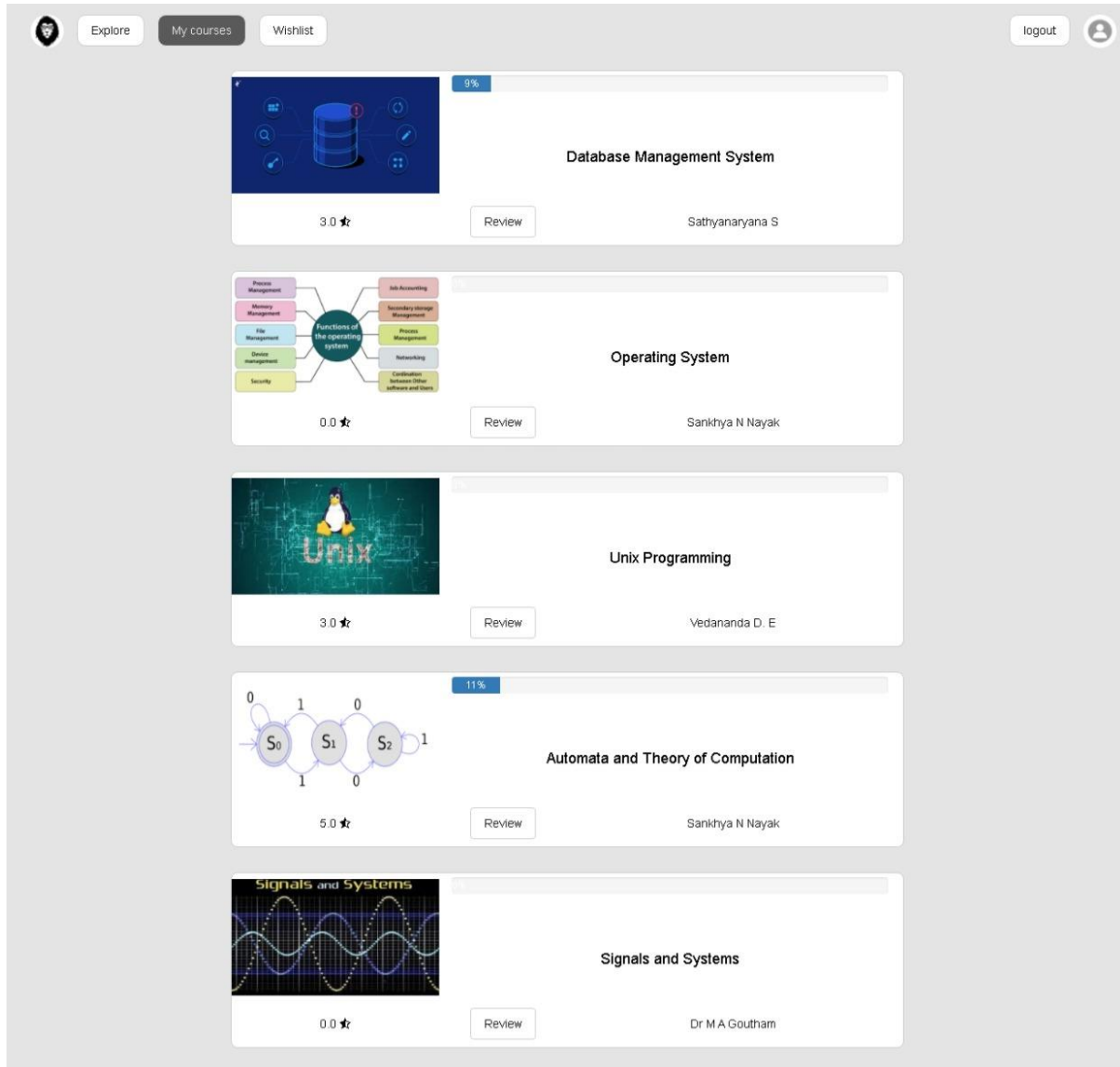
₹850

Guided Project

Figure 3.10



### 3.5.1 My Courses



**Figure 3.11**

**Figure 3.11** shows my courses page of a student. It includes all the courses to which the student has enrolled. The enrollment details are stored in a database table as shown in the **Figure 3.12**. It stores the student id of the student, course id of the course to which he has enrolled and the watched list which contains the indices of the video that has student has watched. My courses page shows the progress of the student in each course with the help of watched list. **Figure 3.13** and **Figure 3.14** shows the course page where video will be played. Right half of the page contains the list of blocks for each video in that course. Each block includes the title, description, thumbnail and duration of the video, and with a icon, it shows whether the video is watched or not.

+ Options




























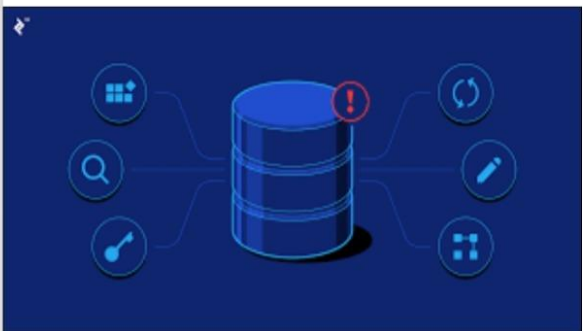
				id	status	c_id_id	s_id_id	watched
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	inprogress	3	3	0,2,4,3,1,5,8,2
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	inprogress	5	2	
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	inprogress	3	2	0,15,1,2
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	inprogress	6	3	
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	inprogress	6	2	
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	inprogress	4	3	0,1,2,7
<input type="checkbox"/>	 Edit	 Copy	 Delete	7	inprogress	7	2	0,1,2,3,4,5
<input type="checkbox"/>	 Edit	 Copy	 Delete	8	inprogress	7	3	
<input type="checkbox"/>	 Edit	 Copy	 Delete	9	inprogress	5	3	

Figure 3.12

back



**Concurrency Control Techniques**  
Transaction and Concurrency Control 0:40:56

**SQL Statements and Concurrency Control**  
Transaction and Concurrency Control 0:45:58

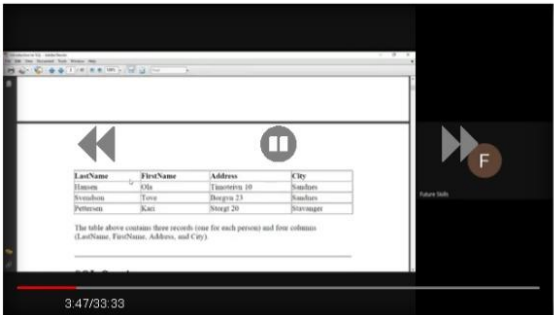
**SQL Statements and Transaction**  
SQL commands and Transaction 0:33:34

**College Database**  
DBMS Lab Assignment 0:43:38

**College Database 1**  
DBMS Lab Assignment 0:55:56

Figure 3.13

back



**SQL Statements and Transaction**  
SQL commands and Transaction

**Concurrency Control Techniques**  
Transaction and Concurrency Control 0:40:56

**SQL Statements and Concurrency Control**  
Transaction and Concurrency Control 0:45:58

**SQL Statements and Transaction**  
SQL commands and Transaction 0:33:34

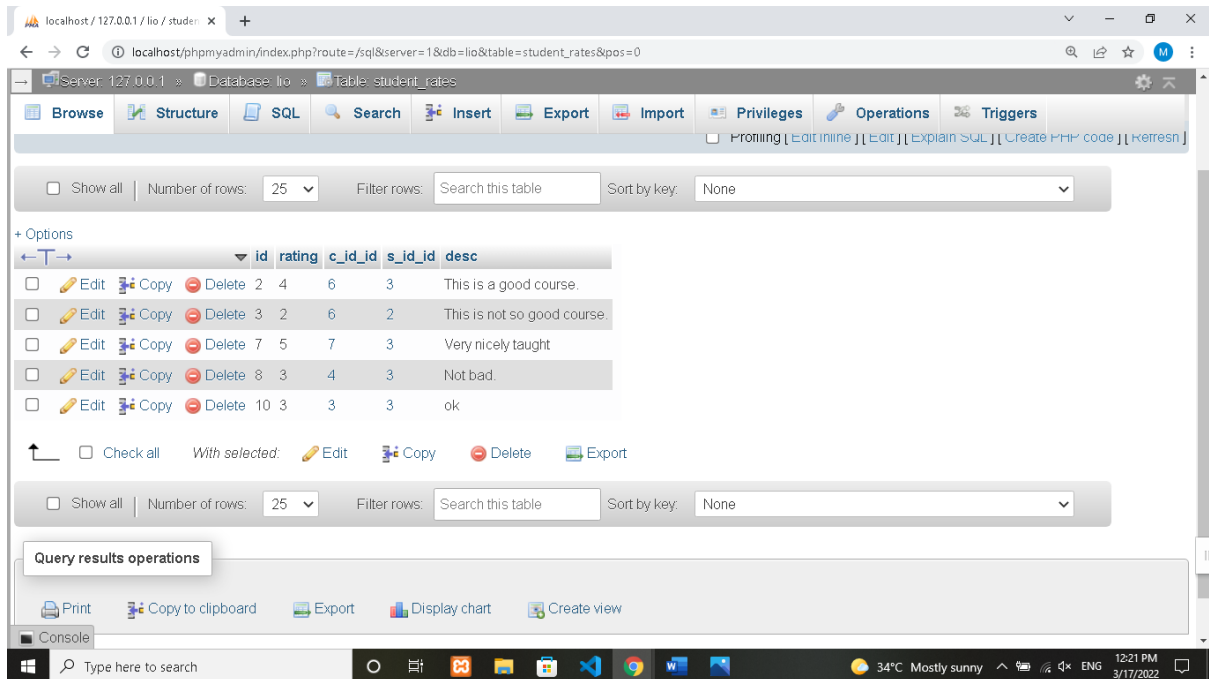
**College Database**  
DBMS Lab Assignment 0:43:38

**College Database 1**  
DBMS Lab Assignment 0:55:56

Figure 3.14



### 3.5.2 Ratings



localhost/127.0.0.1 / io / studen x

localhost/phpmyadmin/index.php?route=/sql&server=1&db=io&table=student\_rates&pos=0

Server: 127.0.0.1 Database: io Table: student\_rates

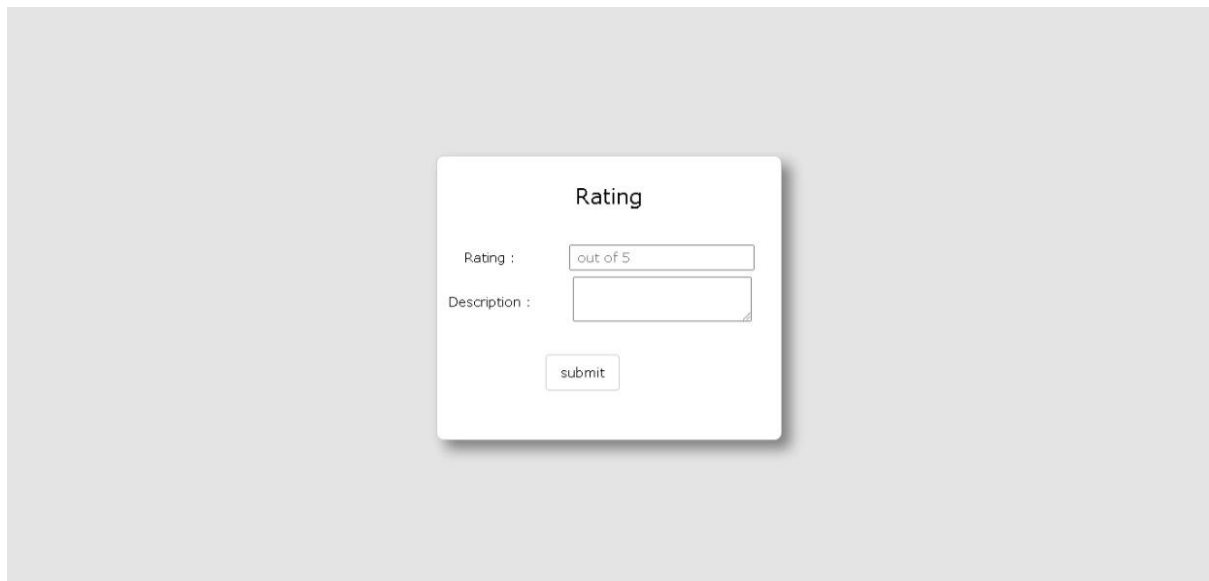
Options: Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	rating	c_id_id	s_id_id	desc
<input type="checkbox"/> Edit Copy Delete	2	4	6	3	This is a good course.
<input type="checkbox"/> Edit Copy Delete	3	2	6	2	This is not so good course.
<input type="checkbox"/> Edit Copy Delete	7	5	7	3	Very nicely taught
<input type="checkbox"/> Edit Copy Delete	8	3	4	3	Not bad.
<input type="checkbox"/> Edit Copy Delete	10	3	3	3	ok

Options: Check all | With selected: Edit Copy Delete Export

Query results operations: Print Copy to clipboard Export Display chart Create view

**Figure 3.15**



### Rating

Rating :

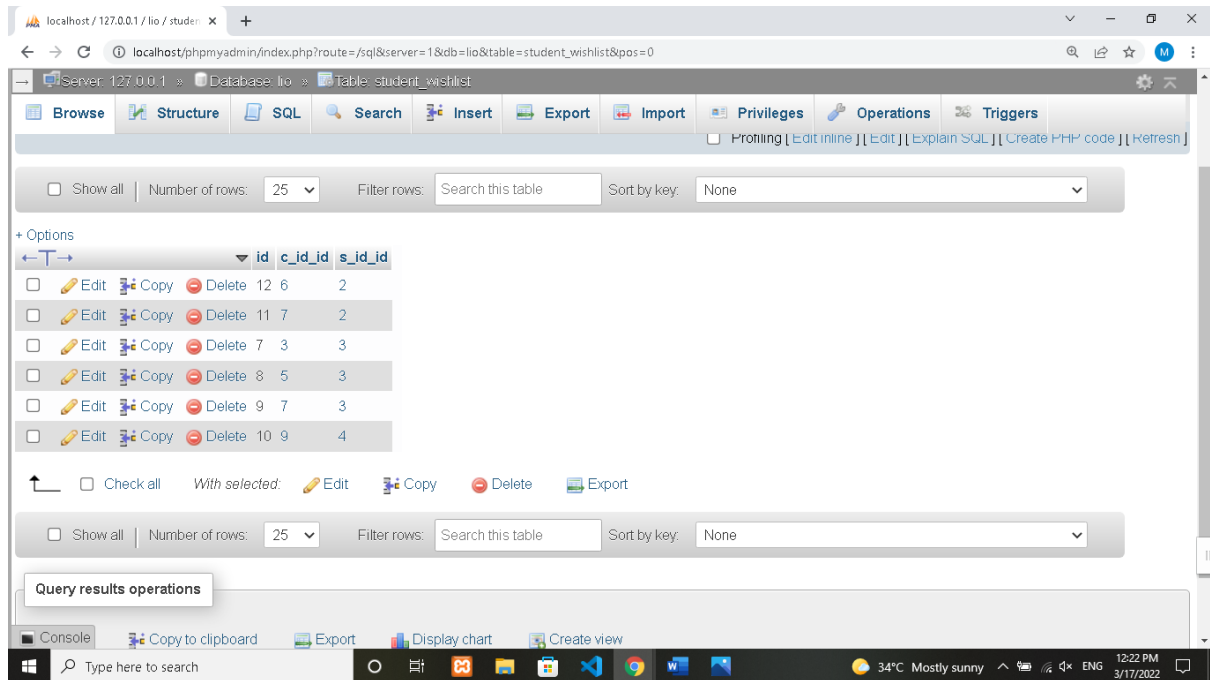
Description :

**Figure 3.16**

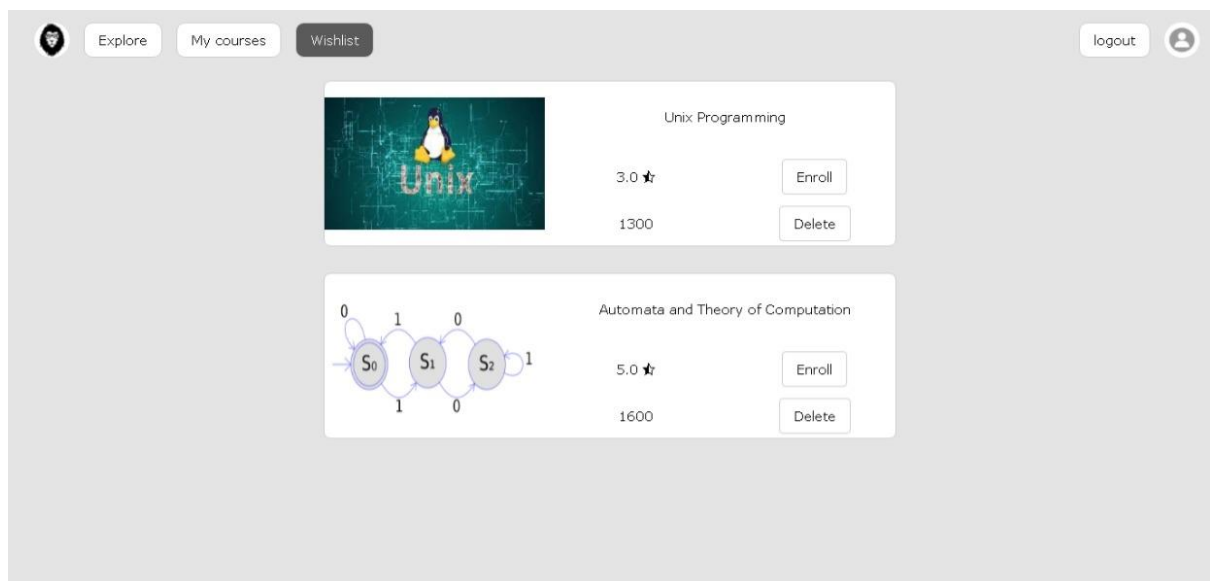
**Figure 3.15** shows the Rating table. This is the table which consists of the rating given by the students to particular course. It stores the student id of the student, course id of the course, and the rating and review the student has given for that course.

**Figure 3.16** shows the rating form which is popped on clicking review button in my courses page. Using this form the student can give the rating. Only enrolled students can give the rating for the course.

### 3.5.3 Wishlist



**Figure 3.17**

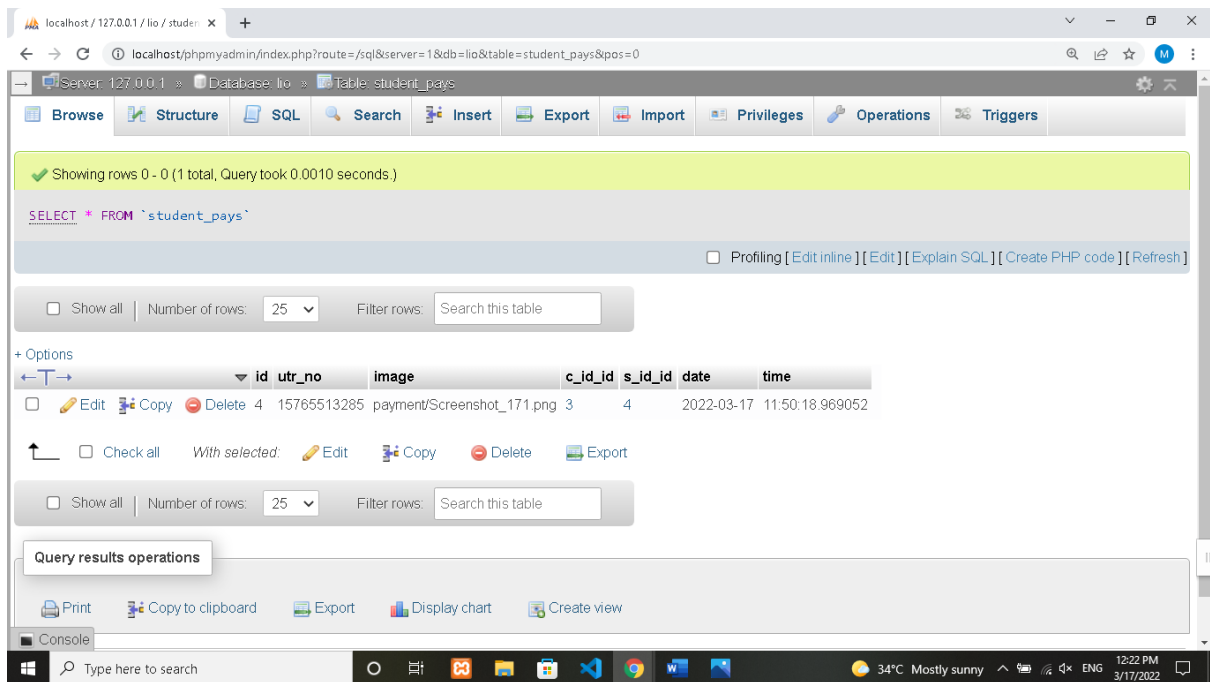


**Figure 3.18**

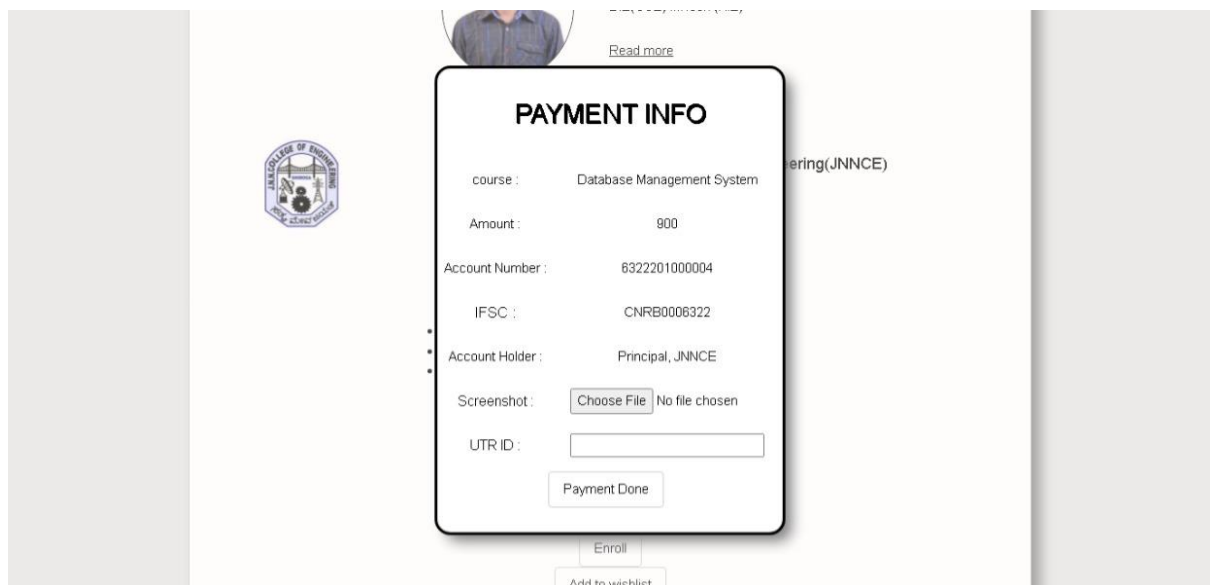
Students can Wishlist courses of which's data will be stored in Wishlist database. id will be given to every course wish listed by student separately and it will act as primary key to select every row of database independently. Data bases consist of id, c\_id\_id, s\_id\_id table which stores id of given to wish listed course, course id and student id respectively

When student login with id, he will be directed to explore page. Students can Wishlist courses which he wants, and it could be seen in Wishlist page. Whenever student want to enroll the course, he will click on to thumbnail and he can enroll.

### 3.5.4 Payment



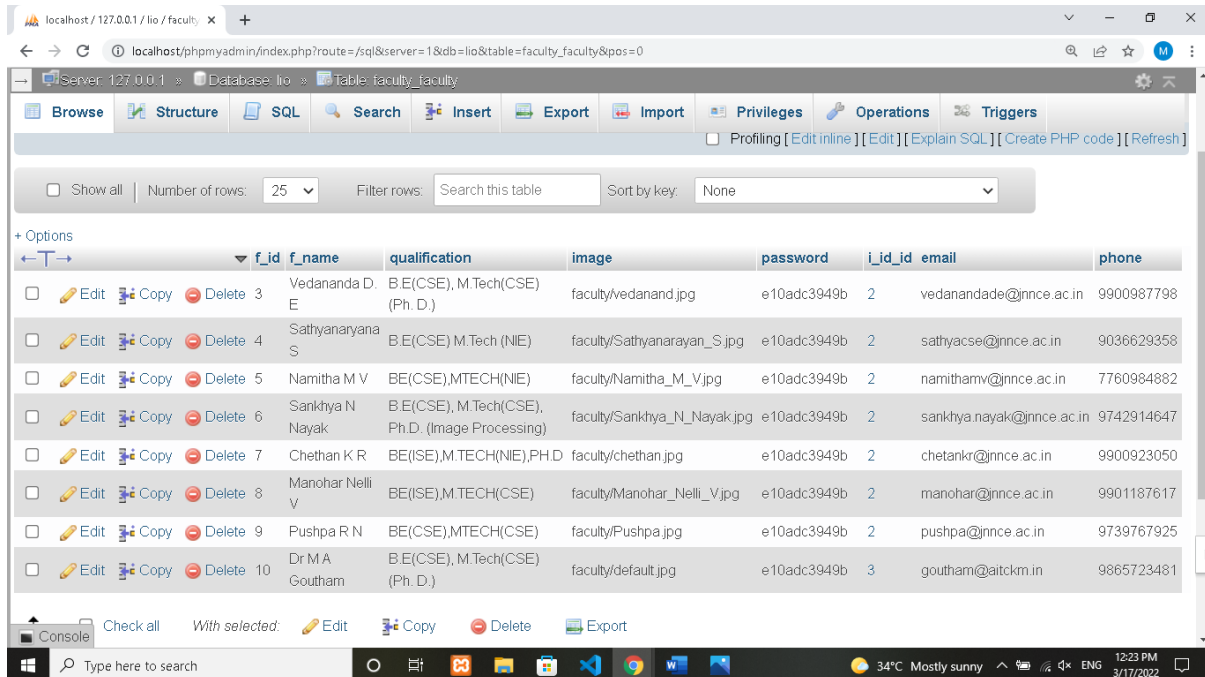
**Figure 3.19**



**Figure 3.20**

Whenever student want to buy some course, he will click on the enroll button and he will be directed to payment form. Where he should upload screen shot and UTR number of transaction and clicks on payment done button to enroll the course.

## 3.6 Faculty



The screenshot shows a web browser window displaying a database management interface. The URL is localhost/phpmyadmin/index.php?route=/sql&server=1&db=lio&table=faculty\_faculty&pos=0. The interface includes tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. Below the tabs, there are options to show all rows, filter rows, and sort by key. The main table displays 10 rows of faculty data with columns: f\_id, f\_name, qualification, image, password, i\_id\_id, email, and phone. Each row has edit, copy, and delete icons. The bottom of the browser shows a Windows taskbar with the date 3/17/2022 and time 12:23 PM.

f_id	f_name	qualification	image	password	i_id_id	email	phone
3	Vedananda D. E	B.E(CSE), M.Tech(CSE) (Ph. D.)	faculty/vedanand.jpg	e10adc3949b	2	vedanandade@jnnce.ac.in	9900987798
4	Sathyanarayana S	B.E(CSE) M.Tech (NIE)	faculty/Sathyanarayan_S.jpg	e10adc3949b	2	sathyacse@jnnce.ac.in	9036629358
5	Namitha M V	BE(CSE),MTECH(NIE)	faculty/Namitha_M_V.jpg	e10adc3949b	2	namithamv@jnnce.ac.in	7760984882
6	Sankhya N Nayak	B.E(CSE), M.Tech(CSE), Ph.D. (Image Processing)	faculty/Sankhya_N_Nayak.jpg	e10adc3949b	2	sankhya.nayak@jnnce.ac.in	9742914647
7	Chethan K R	BE(ISE),M.TECH(NIE),PH.D	faculty/chethan.jpg	e10adc3949b	2	chetankr@jnnce.ac.in	9900923050
8	Manohar Nelli V	BE(ISE),M.TECH(CSE)	faculty/Manohar_Nelli_V.jpg	e10adc3949b	2	manohar@jnnce.ac.in	9901187617
9	Pushpa R N	BE(CSE),MTECH(CSE)	faculty/Pushpa.jpg	e10adc3949b	2	pushpa@jnnce.ac.in	9739767925
10	Dr M A Goutham	B.E(CSE), M.Tech(CSE) (Ph. D.)	faculty/default.jpg	e10adc3949b	3	goutham@aitckm.in	9865723481

**Figure 3.21**

Faculty database will consist of all data including faculty id, faculty name, faculty qualification, image of faculty, password, i\_id\_id, email, phone number of faculty in f\_id, f\_name, qualification, image, password, i\_id\_id, email, phone table.

### 3.6.1 My courses

When lectures login with their IDs. Page will be directed to explore page. When they click on My courses, page My courses will opens which contains all the videos which are uploaded by particular lecturer. There will be upload button. On clicking that, Upload form will opens where user has to give all necessary information and can upload the video.

Hello Sankhya N Nayak

**Operating System** 34:15:19  
Introduce concepts and terminology used in OS. Explain threading and multithreading systems. Illustrate process synchronization and concept of Deadlock. Introduce Memory and Virtual memory management. File systems and storage techniques.  
₹850 0.0 ★ 84 views 3 inprogress 46 videos 0 completed

**Automata and Theory of Computation** 27:57:37  
Introduce core concepts of ATC. Identify different Formal language Classes and their Relationships. Design Grammars and Recognizers for different formal languages. Prove or disprove theorems in automata theory using their properties. Determine the decidability and undecidability of problems.  
₹1800 5.0 ★ 19 views 2 inprogress 52 videos 0 completed

Upload

Figure 3.22

### 3.6.2 Upload

**UPLOAD COURSE**

Title :

image :  No file chosen

specialization :

Forwho :

Playlist\_id :

Type : ☐ Course ☐ Guided Project

description :

skill Acquired :

Price :

Figure 3.23

When lectures login with their IDs and want to upload the video then it will be facilitated by 'upload' page. Where he have to give Title, Thumbnail, specialization, Forwho, playlist id of YouTube, description, skill acquired by course, price.

## 3.7 Institute

The screenshot shows a web application interface for managing a database. The browser address bar indicates the URL is localhost/phpmyadmin/index.php?route=/sql&server=1&db=lio&table=institute\_institute&pos=0. The interface includes a top navigation bar with tabs like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. Below this, a status bar shows 'Showing rows 0 - 1 (2 total, Query took 0.0010 seconds.)'. The SQL query editor contains the query 'SELECT \* FROM `institute\_institute`'. Below the query editor, there are options for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (Search this table), and 'Sort by key' (None). The main table displays two rows of institution data. Each row has a checkbox, an 'Edit' button, a 'Copy' button, and a 'Delete' button. The table columns are: i\_id, i\_name, address, image, email, website, ac\_name, ac\_no, b\_nam, ifsc, and password.

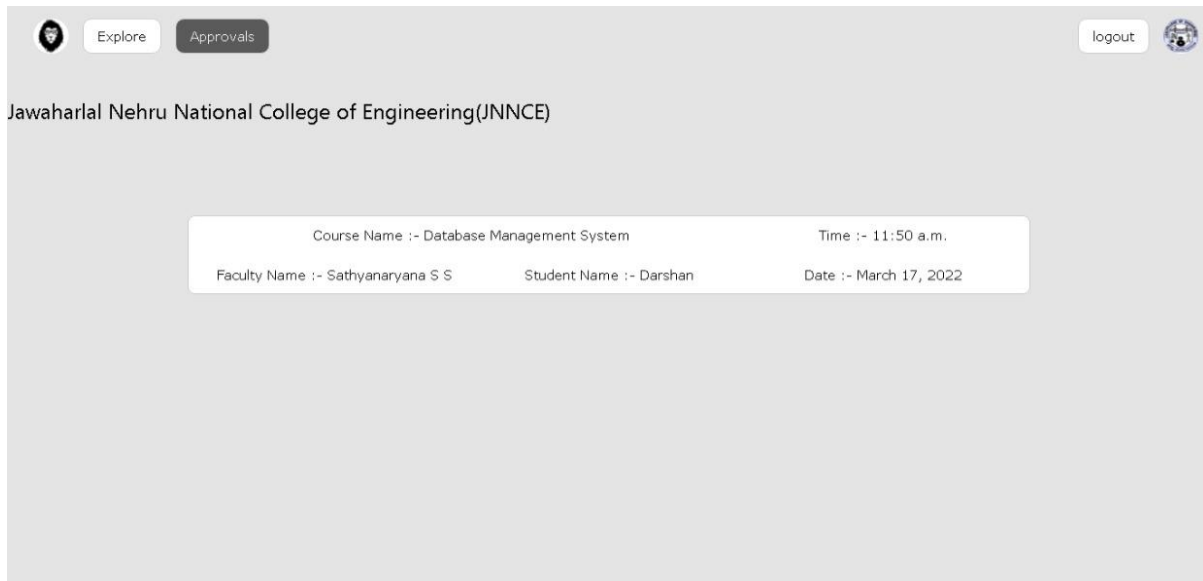
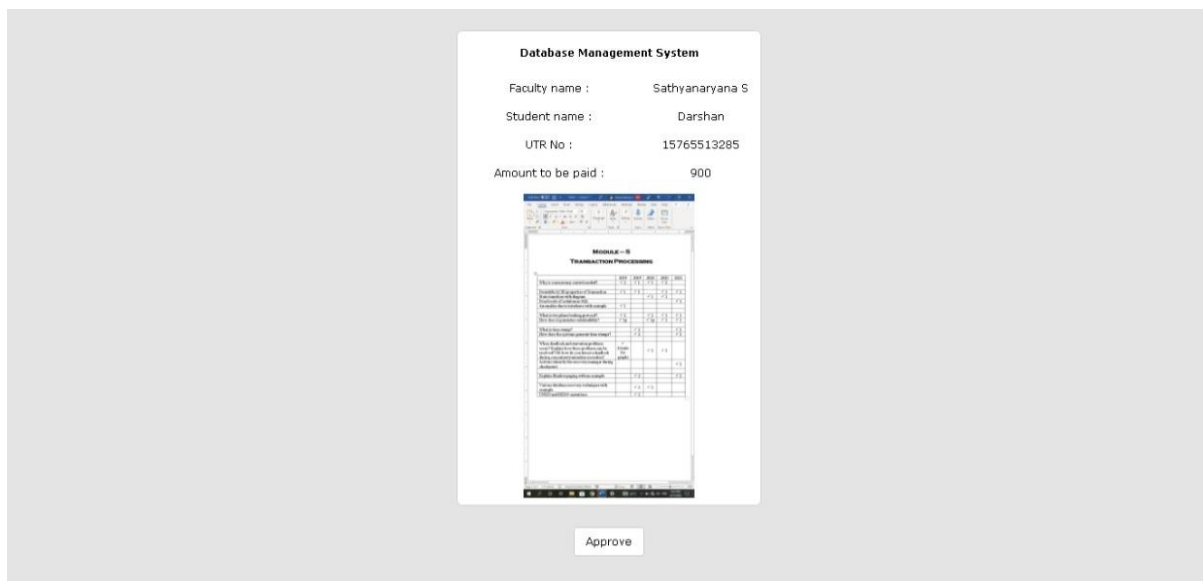
	i_id	i_name	address	image	email	website	ac_name	ac_no	b_nam	ifsc	password
<input type="checkbox"/> Edit Copy Delete	2	Jawaharlal Nehru National College of Engineering(J...	JNNCE, Navule, Shimoga - 577 204, Karnataka, INDIA	institute/jnnc	info@jnnc	http://jnnc.ac.	Principal, JNNCE	6322201C	CANAF BANK	CNRBC	e10adc3949
<input type="checkbox"/> Edit Copy Delete	3	Adichunchanagiri Institute of Technology(AIT)	Adichunchanagiri Institute of Technology, Jyothi n...	institute/AIT	office@aitc	https://aitckm.i	Principal AIT	3705509E	SBI	SBIN0C	e10adc3949

**Figure 3.24**

Data base of institutions which have i\_id, i\_name, address, image of institution, email address, website, ac\_name, ac\_no, b\_nam, ifsc, password. This data table will have information which will be necessary to identify institution, it's courses and to make transactions.

### 3.7.1 Approvals

This page belongs to institutions. When institution login with their IDs, they will get Explore and Approval page. In Approval page they see all the applications of students who want to enroll for particular course. If students are from same institution then they will get some offers otherwise institutions will have approve explicitly.

**Figure 3.25****Figure 3.26**

### 3.8 Form Errors

The screenshot shows a web application interface with a modal form titled "INSTITUTE REGISTER". The form contains several input fields: Institute, Address, Image, Email, Website, Bank Name, Account Holder, Account Number, IFSC Code, Password, and confirm password. A red error block is overlaid on the form, displaying the following messages:

- ERROR**
- Institute :** This field is required.
- Address :** This field is required.
- Image :** This field is required.
- Email :** This field is required.
- Website :** This field is required.
- Bank Name :** This field is required.
- Account Holder :** This field is required.
- Account Number :** This field is required.
- IFSC Code :** This field is required.
- Password :** This field is required.

The background of the application shows a navigation bar with "Explore", "login", and a user profile icon. Below the navigation bar are category buttons for "Web development", "Java", "Computer Science", and "In". The main content area features a "Most Viewed" section with course cards for "Unix Programming" (Intermediate, 28:24:1, ₹1300, 3.0 stars) and "Operating Systems" (Intermediate, ₹850). To the right, there are more course cards for "Microcontroller and Embedded Systems" (Beginners, 12:3:30, ₹1100, 0.0 stars) and "Automata and Theory of Computation" (Advanced, 27:16:00, ₹1600, 5.0 stars).

Figure 3.27

If the user commits any error while filling the form, then the error block will be displayed after submitting that form as shown in the above figure.



## **CHAPTER – 4**

### **CONCLUSION AND FUTURE SCOPE**

The problem in live class is that there is no guaranteed bandwidth which causes packet loss and this effects the continuity of the class. But in online learning as our project, videos are prerecorded and available at any time. Therefore, it is time flexible. Since the videos are prerecorded, if there is any packet loss then they can just go back few seconds in the video. And these courses are self-paced. Once the student is enrolled for a course then he can watch the videos any number of times.

#### **Future Scope:**

- A feature for users to change their passwords, image, phone number.
- A feature for faculty to update the title, description, skills, price, level (for who). And a feature to update the number of videos and duration if they have added or removed a video from the playlist.
- A feature where the faculty can add assignments while uploading the course or can add it by updating an existing course.
- A feature where the institute has to verify that the faculty or student who claims to work or study in their institute respectively is true.
- A feature where the institute can provide the certificate to the students who completes the course.

## References: -

- [1] Antonio Mele, Django3 by Example, 3<sup>rd</sup> Edition, Pack Publishers, 2020.
- [2] Daniel Rubio, Beginning Django: Web Application Development and Deployment with Python, APress, 1stEdition, 2017.
- [3] Arun Ravindran, Django Design Patterns and Best Practices, 2<sup>nd</sup> Edition, Pack Publishers, 2020.
- [4] Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Publications, 2016
- [5] <https://erdplus.com/>
- [6] <https://www.w3schools.com/w3css/>
- [7] Vikram Vaswani, MYSQL The Complete Reference, Tata McgrawHill, 1<sup>st</sup> Edition, 2017.