An

INDUSTRIAL TRAINING REPORT

on

INDUSTRY MITRA

at

DIGIDIYA

Final Semester Training

Submitted for the partial fulfilment for the award of Degree of

Bachelor of Technology

in

Computer Science & Engineering

J.C. BOSE UNIVERSITY OF SCIENCE & TECHNOLOGY, (YMCA)

FARIDABAD

Session (2017-2021)

Submitted By

Gaurav Kataria

(17020004022)

Under the Guidance of

Mr Akash Malik

(External Project Guide)

MR KAUSHAL KUMAR

(Head of Department)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATYUG DARSHAN INSTITUTE OF ENGINEERING & TECHNOLOGY

FARIDABAD

DigiDiya

Enlighten Your Growth

June 21, 2021

To WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Gaurav Kataria** Candidate ID: 121 associated with us from 01.01.2021 to 30.04.2021. His designation was **Angular Developer Intern.**

During this period his conduct was good. He has no dues left in the company

We wish him all the best for future endeavors.

Warm Regards,
HRD Team
Digidiya

+91-8813939898
+91-9540672377
www.digidiya.com
info@digidiya.com
Office No. 4, B-88, Sector 2,
Noida - 201301, U.P. INDIA

DECLARATION

I hereby certify that the work which is being presented in this project report titled **"Industry Mitra"** in fulfilment of the requirement for the degree of Bachelor of Technology and submitted to "**J. C. Bose University of Science and Technology, YMCA, Faridabad**", is an authentic record of my own work carried out under the supervision of **Mr. Akash Malik.**

The work contained in this report has not been submitted to any other University or Institute for the award of any other degree or diploma by me.

Gaurav Kataria

17020004022

ACKNOWLEDGEMENT

The successful completion of this internship marks the beginning of an ever - going learning experience of converting ideas and concepts into real life, practical systems. This project was quite a learning experience for me at each and every step. At the same time it has given me confidence to work in a professional setup. I feel the experience gained during the project will lead me to gain a bright prospect in the future. First of all I would like to give thanks to **Mr Akash Malik** for giving me the opportunity to work in this esteemed organization, which not only has increased our awareness about latest fields but also taught me the importance of team building. With a deep sense of gratitude, I express my sincere thanks to the Head of CSE Department ,**Mr. Kaushal Kumar,** for his active support and continuous guidance without which it would have been difficult for me to complete this project.

Gaurav Kataria

17020004022

TABLE OF CONTENTS

Chapter 1

COMPANY PROFILE

COMPANY PROFILE

DIGIDIYA is one of the most rapidly growing online solutions companies in India, providing an array of web design, web development, graphic design and digital marketing solutions to an esteemed global clientele. Business profits for you; our biggest solution for you. No matter what, we never forget one thing. Business is all about profits. Whatever work we do for you is designed to provide you with the profits you seek for your business. Work with us, and we'll guarantee you profits. Our specializations are vast and we are able to draw on a wealth of expertise to ensure that we provide you with the profits that you are looking for. Our updated infrastructure coupled with our team strength of professionals enables us to blend superior technology with core functional expertise to deliver outstanding solutions

1.1 COMPANY WORK SCOPE

1. Web Design and development
2. Digital Marketing
3. Android and IOS application development
4. SEO.
5. Traffic analysis

CHAPTER 2

INTRODUCTION OF PROJECT

INTRODUCTION OF PROJECT

While working in Digi Diya, I was tied up with a client **Piruby,** they have many projects on which I worked. But the main of them was **Industry Mitra.** Industry Mitra is basically a learning platform on which students can find different courses, on which they can watch videos, give different types of quizzes, ask questions to faculty and many other things. There are many learning platforms like this in the market, but the main thing that makes Industry Mitra unique is, the platform observes all the activities of the student, like when someone is watching a video when someone is taking class and many other things

Some Features in this platform are listed below: -

1. Students can attend class and faculty can see for how much time a student is active.
2. Students can take quizzes and faculty can see how students are performing in real time.
3. Students can see different types of videos and their watch time is getting stored.
4. Students can solve exam papers and upload their solutions and faculty can give them scores.

Different types of roles in this platform are: -

1. Admin: - Admins can add different courses, categories, users to this application. There main purpose is to maintain the website and update the data according to different scenarios
2. Faculty: - Whenever a course is published, a faculty is assigned to that course, and that faculty can add different type of content to that course like, videos, labs, quizzes, etc
3. Student: - Students are end users of this platform. They can purchase different courses and take part in them by joining classes and viewing content.

Chapter 3

REQUIREMENT ANALYSIS

REQUIREMENT ANALYSIS

3.1 HARDWARE REQUIREMENTS

- PROCESSOR: Pentium IV processor or Greater
- RAM: 128 Mega Byte (MB) or Greater
- HARDDISK : 1.2 Giga Byte (GB) or Greater
- Keyboard & Mouse
- MONITOR : Colour (For Best Result)

3.2 SOFTWARE REQUIREMENTS

3.2.1 Tools Used

1. Vs code: -

Visual Studio Code is a distribution of the Code - OSS repository with Microsoft specific customizations released under a traditional Microsoft product license.

Visual Studio Code combines the simplicity of a code editor with what developers need for their core edit-build-debug cycle. It provides comprehensive code editing, navigation, and understanding support along with lightweight debugging, a rich extensibility model, and lightweight integration with existing tools.

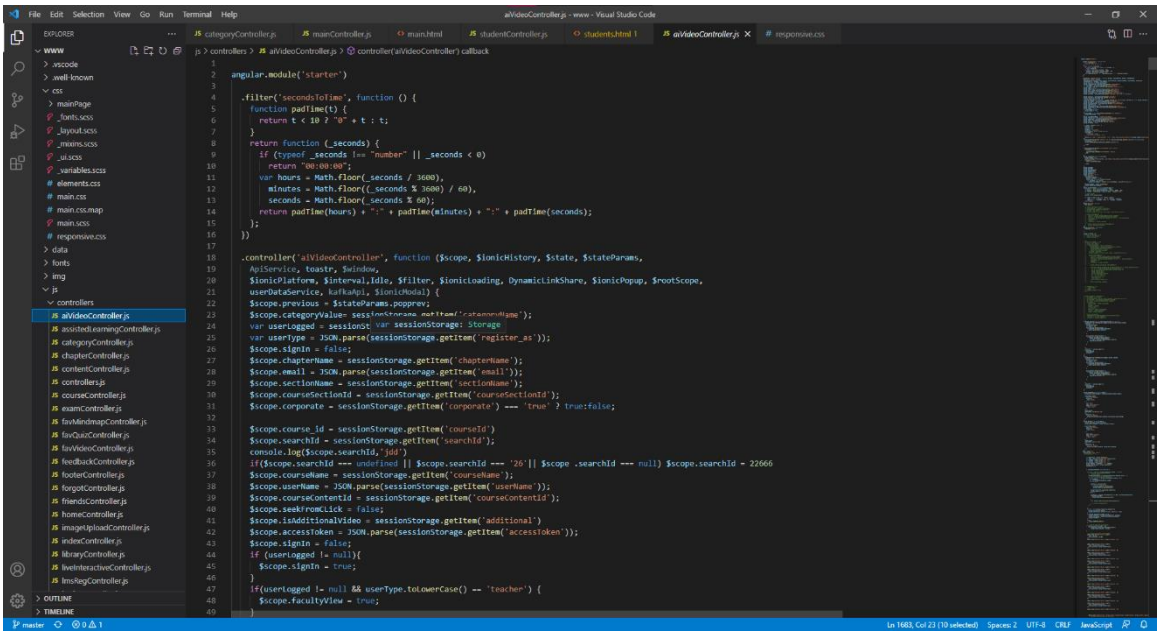

Fig 3.2.1 Vs Code GUI

2. Google Chrome Dev tools: -

Google chrome developer tools are tools which help a developer to develop ,debug and analyse the website

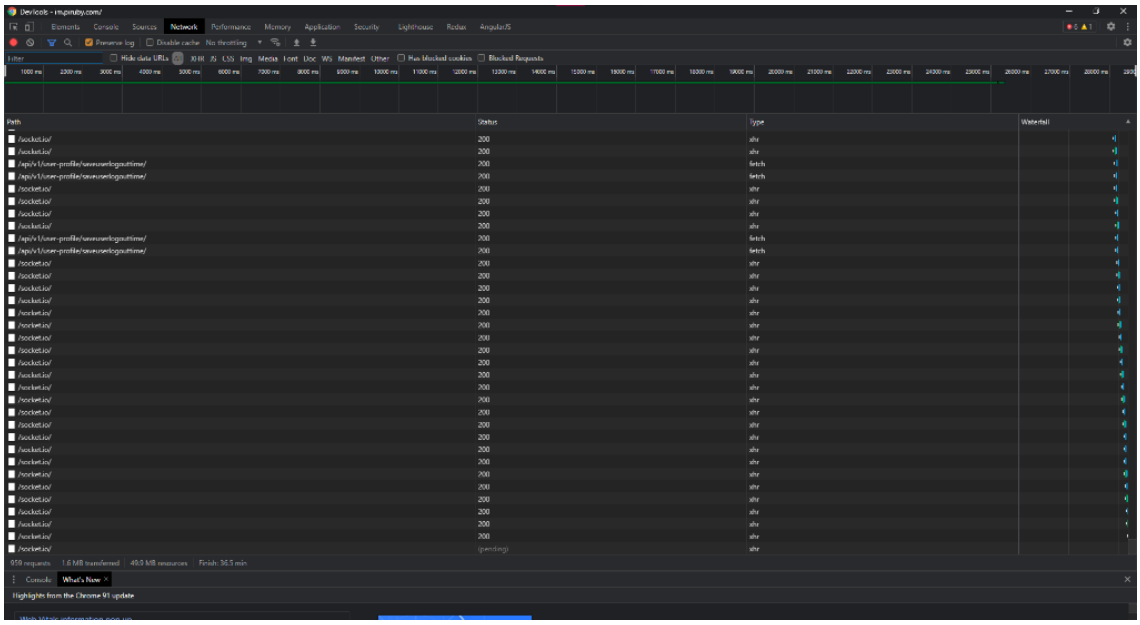

Fig 3.2.2 Google Chrome Dev tools

3. SSH: -

SSH, also known as Secure Shell or Secure Socket Shell, is a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network. In addition to providing secure network services, SSH refers to the suite of utilities that implement the SSH protocol. Secure Shell provides strong password authentication and public key authentication, as well as encrypted data communications between two computers connecting over an open network, such as the internet. In addition to providing strong encryption, SSH is widely used by network administrators for managing systems and applications remotely, enabling them to log in to another computer over a network, execute commands and move files from one computer to another.

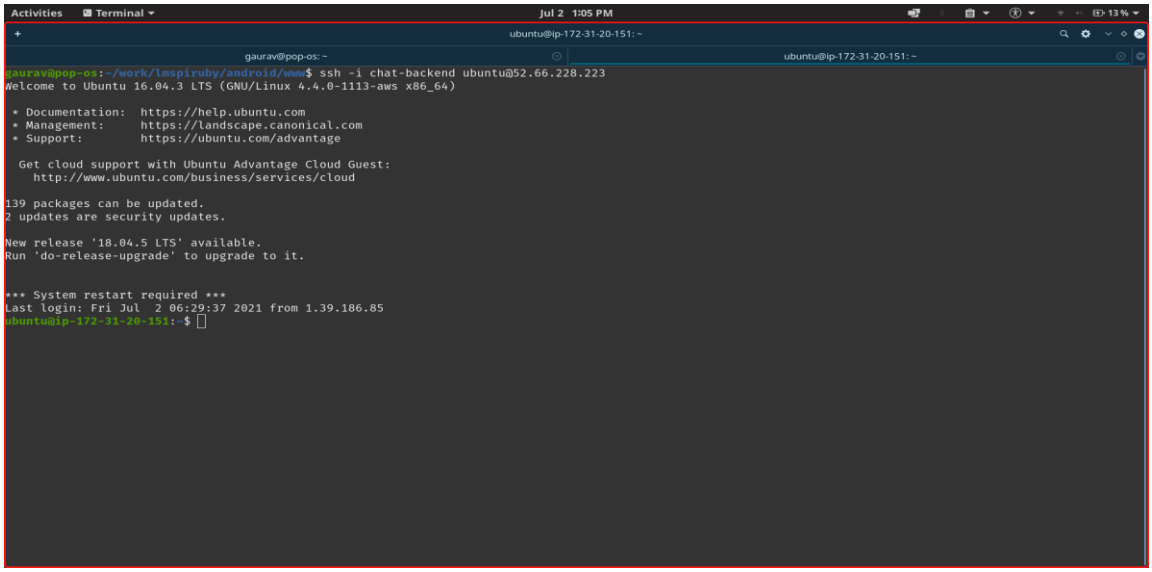

Fig 3.2.3 Doing SSH in terminal

4. Git: -

Git is a version control system that developers use all over the world. It helps you track different versions of your code and collaborate with other developers.

If you are working on a project over time, you may want to keep track of which changes were made, by whom, and when those changes were made. This becomes increasingly important if you end up having a bug in your code! Git can help you with this.

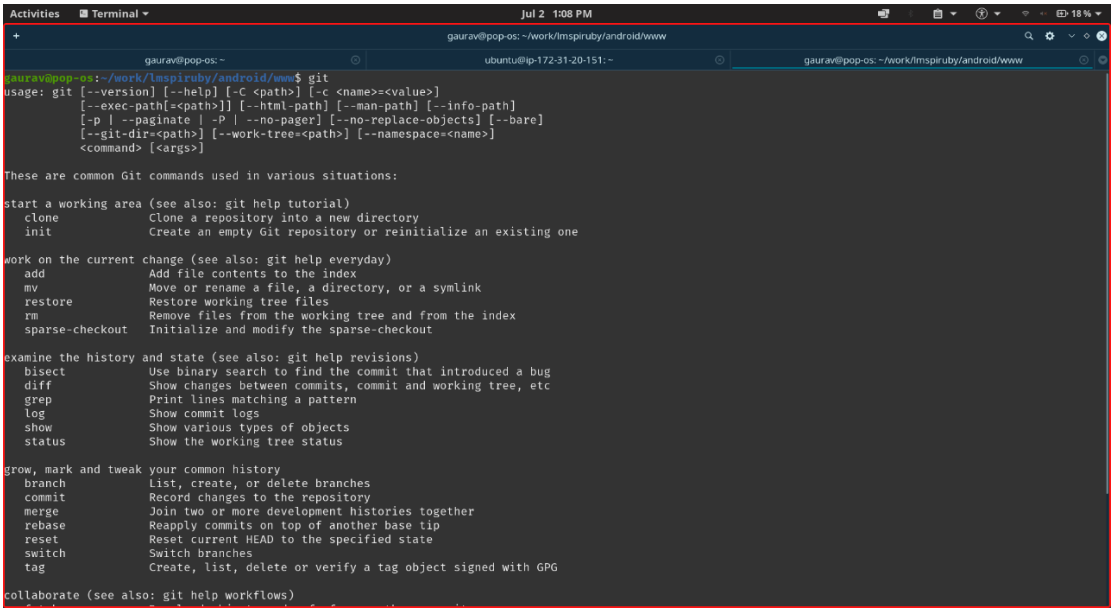

Fig 3.2.4 Using Git in terminal

3.2.2 Software Technologies Used

3.2.2.1 Introduction to Node js

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project!Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant. A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behaviour the exception rather than the norm.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back. This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

Features of Node.js

Following is some of the important features that make Node.js the first choice of software architects.

- Asynchronous and Event Driven − All APIs of Node.js library is asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- Very Fast − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- Single Threaded but Highly Scalable − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional

servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

- No Buffering − Node.js applications never buffer any data. These applications simply output the data in chunks.
- License − Node.js is released under the MIT license

Concepts

The following diagram depicts some important parts of Node.js which we will discuss in detail in the subsequent chapters.

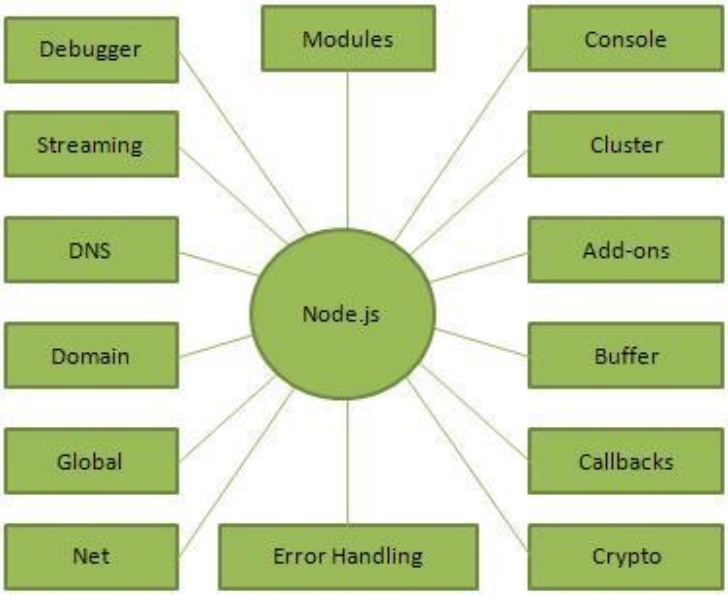

Fig 3.2.5 Node js Concepts

Where to Use Node.js?

Following are the areas where Node.js is proving itself as a perfect technology partner.

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

Where not to Use Node.js?

It is not advisable to use Node.js for CPU intensive applications.

3.3.2 Introduction to Angular Js

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. AngularJS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology. AngularJS is not a single piece in the overall puzzle of building the client-side of a web application. It handles all of the DOM and AJAX glue code you once wrote by hand and puts it in a well-defined structure. This makes AngularJS opinionated about how a CRUD (Create, Read, Update, Delete) application should be built. But while it is opinionated, it also tries to make sure that its opinion is just a starting point you can easily change. AngularJS comes with the following out-of-the-box:

- Everything you need to build a CRUD app in a cohesive set: Data-binding, basic templating directives, form validation, routing, deep-linking, reusable components and dependency injection.
- Testability story: Unit-testing, end-to-end testing, mocks and test harnesses.
- Seed application with directory layout and test scripts as a starting point.

General Features

The general features of AngularJS are as follows −

- AngularJS is an efficient framework that can create Rich Internet Applications (RIA).
- AngularJS provides developers an option to write client-side applications using JavaScript in a clean Model View Controller (MVC) way.
- Applications written in AngularJS are cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.
- AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache license version 2.0.

Core Features

The core features of AngularJS are as follows −

- Data-binding − It is the automatic synchronization of data between model and view components.
- Scope − These are objects that refer to the model. They act as a glue between controller and view.
- Controller − These are JavaScript functions bound to a particular scope.
- Services − AngularJS comes with several built-in services such as $http to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.
- Filters − These select a subset of items from an array and returns a new array.
- Directives − Directives are markers on DOM elements such as elements, attributes, css, and more. These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives such as ngBind, ngModel, etc.
- Templates − These are the rendered view with information from the controller and model. These can be a single file (such as index.html) or multiple views in one page using *partials*.
- Routing − It is concept of switching views.
- Model View Whatever − MVW is a design pattern for dividing an application into different parts called Model, View, and Controller, each with distinct responsibilities. AngularJS does not implement MVC in the traditional sense, but rather something closer to MVVM (Model-View-ViewModel). The Angular JS team refers it humorously as Model View Whatever.
- Deep Linking − Deep linking allows to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.
- Dependency Injection − AngularJS has a built-in dependency injection subsystem that helps the developer to create, understand, and test the applications easily.

Concepts

The following diagram depicts some important parts of AngularJS which we will discuss in detail in the subsequent chapters.

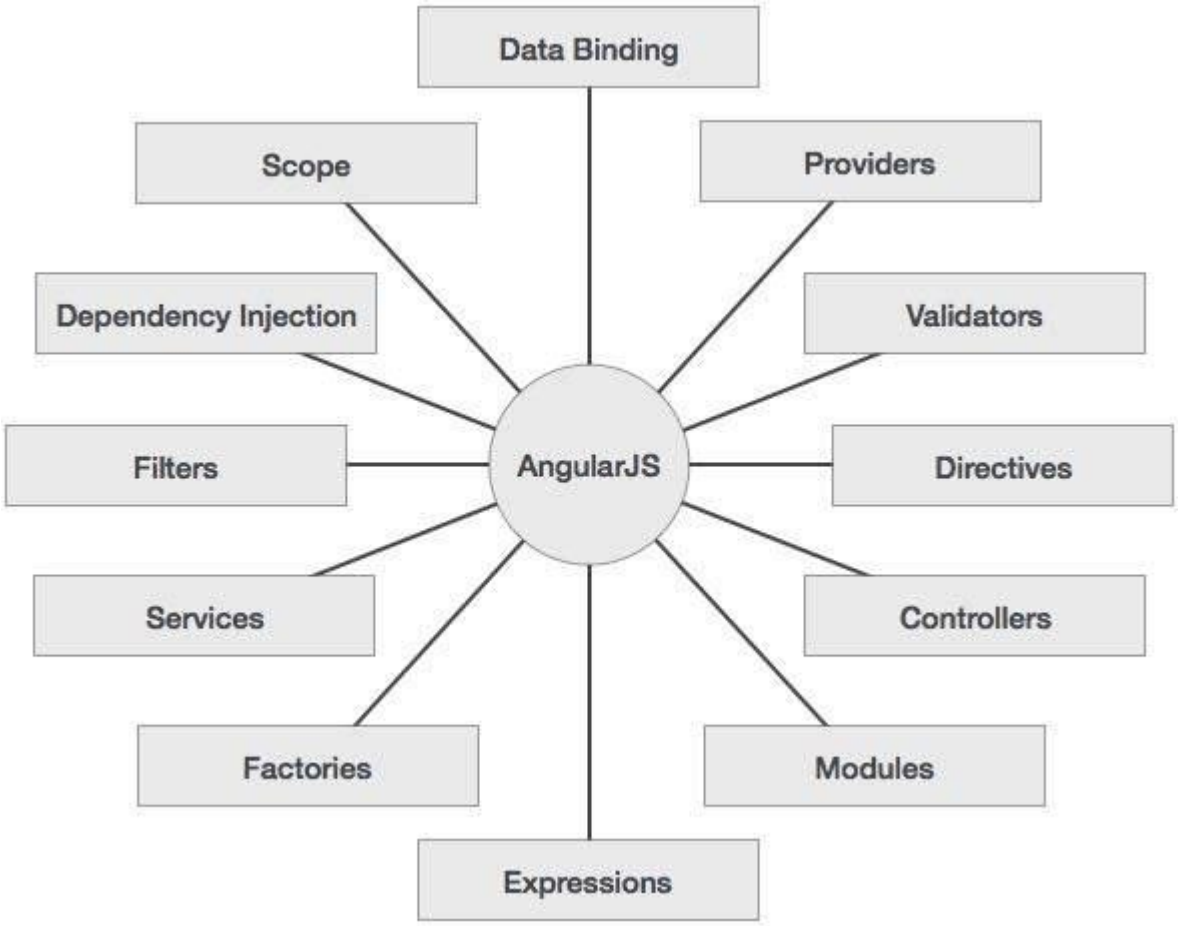

Fig 3.2.6 Angular JS Concepts

Advantages of AngularJS

The advantages of AngularJS are −

- It provides the capability to create Single Page Application in a very clean and maintainable way.
- It provides data binding capability to HTML. Thus, it gives user a rich and responsive experience.
- AngularJS code is unit testable.
- AngularJS uses dependency injection and make use of separation of concerns.
- AngularJS provides reusable components.

- With AngularJS, the developers can achieve more functionality with short code.
- In AngularJS, views are pure html pages, and controllers written in JavaScript do the business processing.

On the top of everything, AngularJS applications can run on all major browsers and smartphones, including Android and iOS based phones/tablets.

Disadvantages of AngularJS

Though AngularJS comes with a lot of merits, here are some points of concern −

- Not Secure − Being JavaScript only framework, application written in AngularJS are not safe. Server side authentication and authorization is must to keep an application secure.
- Not degradable − If the user of your application disables JavaScript, then nothing would be visible, except the basic page.

AngularJS Directives

The AngularJS framework can be divided into three major parts −

- ng-app − This directive defines and links an AngularJS application to HTML.
- ng-model − This directive binds the values of AngularJS application data to HTML input controls.
- ng-bind − This directive binds the AngularJS application data to HTML tags.

3.3.3 Introduction to My SQL

Structure Query Language(SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS(MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

What is a Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would

not be so fast and easy with those type of systems. Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys.

A Relational Database Management System (RDBMS) is a software that −

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

RDBMS Terminology

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

- Database − A database is a collection of tables, with related data.
- Table − A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- Column − One column (data element) contains data of one and the same kind, for example the column postcode.
- Row − A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- Redundancy − Storing data twice, redundantly to make the system faster.
- Primary Key − A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
- Foreign Key − A foreign key is the linking pin between two tables.
- Compound Key − A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- Index − An index in a database resembles an index at the back of a book.
- Referential Integrity − Referential Integrity makes sure that a foreign key value always points to an existing row.

Features of SQL

1. Easy to use: -

   MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.

2. It is secure: -

   MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

3. Client/ Server Architecture: -

   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.

4. Free to download: -

   MySQL is free to use so that we can download it from MySQL official website without any cost.

5. It is scalable: -

   MySQL supports multi-threading that makes it easily scalable. It can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, we can increase this number to a theoretical limit of 8 TB of data.

6. Speed: -

   MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.

7. High Flexibility: -

   MySQL supports a large number of embedded applications, which makes MySQL very flexible.

8. Compatible on many operating systems: -

   MySQL is compatible to run on many operating systems, like Novell NetWare, Windows* Linux*, many varieties of UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others. MySQL also provides a facility that

the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

9. Allows roll-back: -

MySQL allows transactions to be rolled back, commit, and crash recovery.

10. Memory efficiency: -

Its efficiency is high because it has a very low memory leakage problem.

11. High Performance: -

MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.

12. High Productivity: -

MySQL uses Triggers, Stored procedures, and views that allow the developer to give higher productivity.

13. Platform Independent: -

It can download, install, and execute on most of the available operating systems.

14. Partitioning: -

This feature improves the performance and provides fast management of the large database.

15. GUI Support: -

MySQL provides a unified visual database graphical user interface tool named "**MySQL Workbench**" to work with database architects, developers, and Database Administrators. MySQL Workbench provides SQL development, data modelling, data migration, and comprehensive administration tools for server configuration, user administration, backup, and many more. MySQL has a fully GUI supports from MySQL Server version 5.6 and higher.

16. Dual Password Support: -

MySQL version 8.0 provides support for dual passwords: one is the current password, and another is a secondary password, which allows us to transition to the new password.

Disadvantages/Drawback of MySQL

Following are the few disadvantages of MySQL:

- MySQL version less than 5.0 doesn't support ROLE, COMMIT, and stored procedure.
- MySQL does not support a very large database size as efficiently.
- MySQL doesn't handle transactions very efficiently, and it is prone to data corruption.
- MySQL is accused that it doesn't have a good developing and debugging tool compared to paid databases.
- MySQL doesn't support SQL check constraints.

3.3.4 Introduction to AWS Ec2 Instance

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic. Some features of EC2 are

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)

- Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*

Chapter 4

DESIGN

DESIGN

4.1 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Fig 4.1.1 Use Case Diagram for Admin

Fig 4.1.2 Use Case Diagram for Student

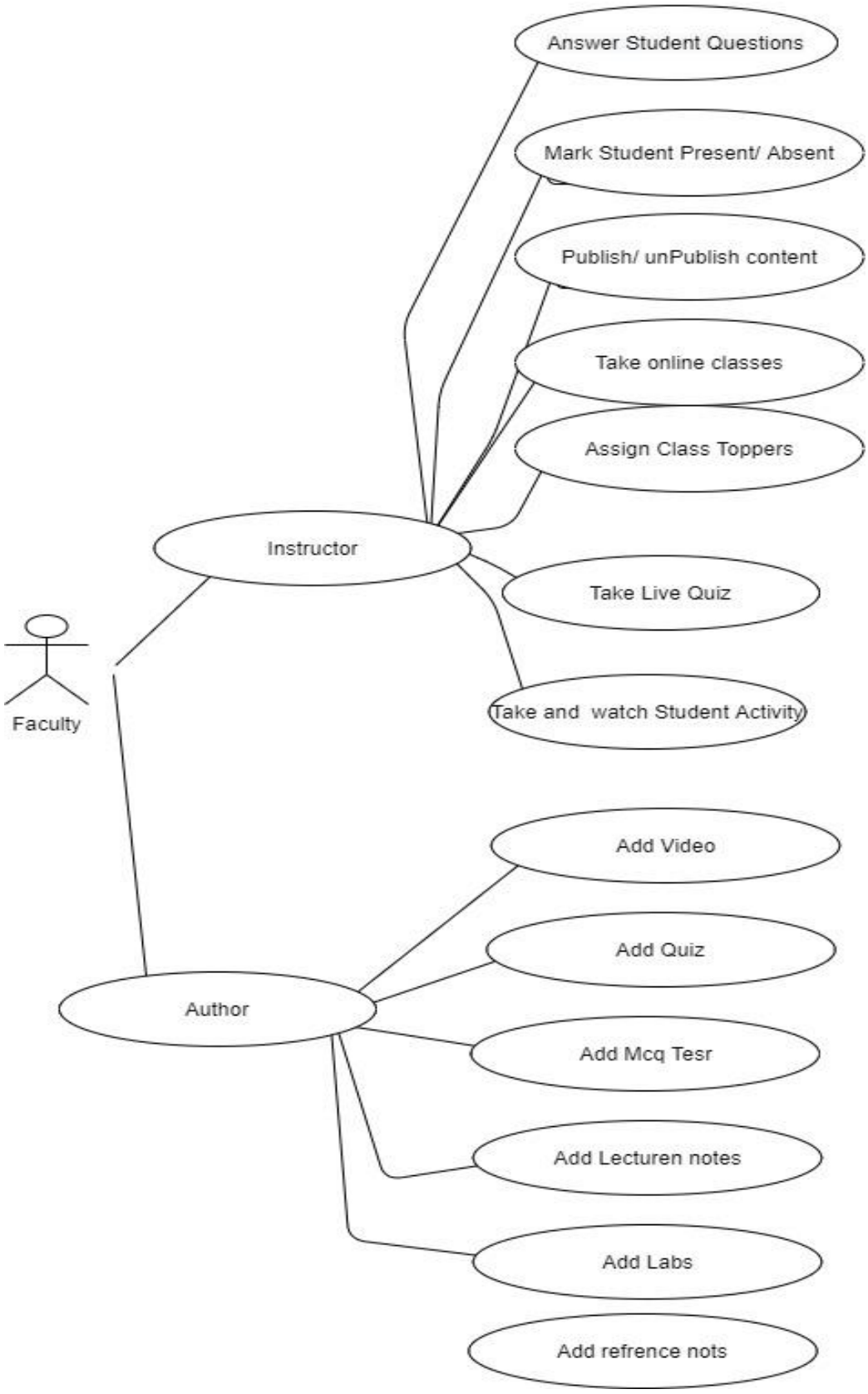

Fig 4.1.3 Use Case Diagram for Faculty

4.2 Level 0 DFD

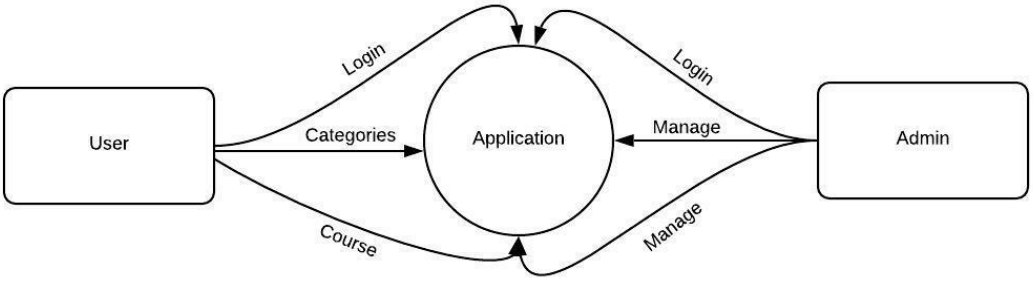

Fig 4.2.1 Level 0 DFD

4.3 Level 1 DFD

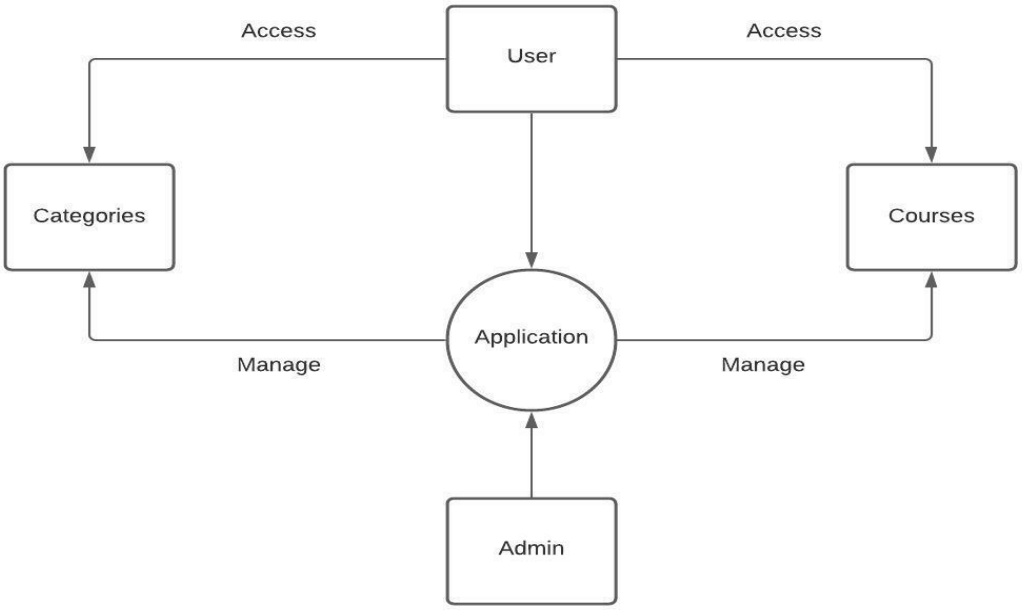

Fig 4.3.1 Level 1 DFD

Chapter 5

IMPLEMENTATION

IMPLEMENTATION

5.1 ATTENDANCE MODULE

In this platform faculty can schedule different classes on different days of a week , according to the schedule time faculty has the option to start the class and when faculty start the class , a student of that particular course can join the class and his class joining activity will shown to the faculty in form of a table and every 15 minutes in the class time , a popup will come to the student with a notification and notification sound to confirm that student is still active and all this data can be shown by the faculty in form of green and red bars where red means absent and green means present. Faculty can also check the cumulative report for a specific period and also download all these data in an excel file.

5.2 TOPPERS MODULE

According to a student performance the system will recommend some students as topper to faculty , A faculty can approve those students as a topper and there names will come in toppers section of the course page , In addition to that , a faculty can give the topper award button to any student of the course with different title. Toppers will get reset after every week

5.3 ADMIN MODULE

In this mode we can add and edit different course , add different categories , add users to different courses , give different permission to faculty , give users permission to access a course , publish and unpublish a course and many other features.

5.4 COURSE MODE MODULE:

There are basically 3 modes of a course, all the modes have different features and functionality in it

- Live Mode - As the name suggests this mode is for courses which are in live mode for direct faculty and student interaction.
- Self-Learning Mode – In this mode there is no faculty involved, student can come, see the content, complete the course and download his certificate.
- Self-Learning Plus mode - In this mode in addition to normal content one lab feature is there, in labs he can perform the skill he learned in the course and ask faculty there doubts and questions for a better understanding

5.5 CHAT MODULE

This feature I s used by different students and faculty to chat with each other between and after the class time, to clear their doubts and ask and discuss different things.
This feature uses the popular socket library which helps to create connection between different users.

5.6 RATING MODULE

Course students can basically give rating to course faculty on a weekly basis. The rating can be given between 1 - 5 stars.

5.7 TOPICS MODULE

Course contains different topics and subtopics , in specific order , a author can easily edit and add topics and subtopics and also can change the order of topic and subtopic.

5.8 QUIZ MODULE

We can take Quiz in two different modes

- Live mode
- Mcq mode

Live Mode :- In live mode faculty has to start a quiz then student can answer questions but he cannot navigate to different questions he has to attempt questions serially. Faculty can see the answers given by the student.

MCQ Mode: - In MCQ mode faculty has to start a quiz with a specified time block then student can start MCQ in that time block and he can navigate between the questions, The questions in mcq mode are shuffled for every student . Students can check their results once the faculty publish the results.

Chapter 6

DATABASE

DATABASE

6.1 DATABASE SCHEMA

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

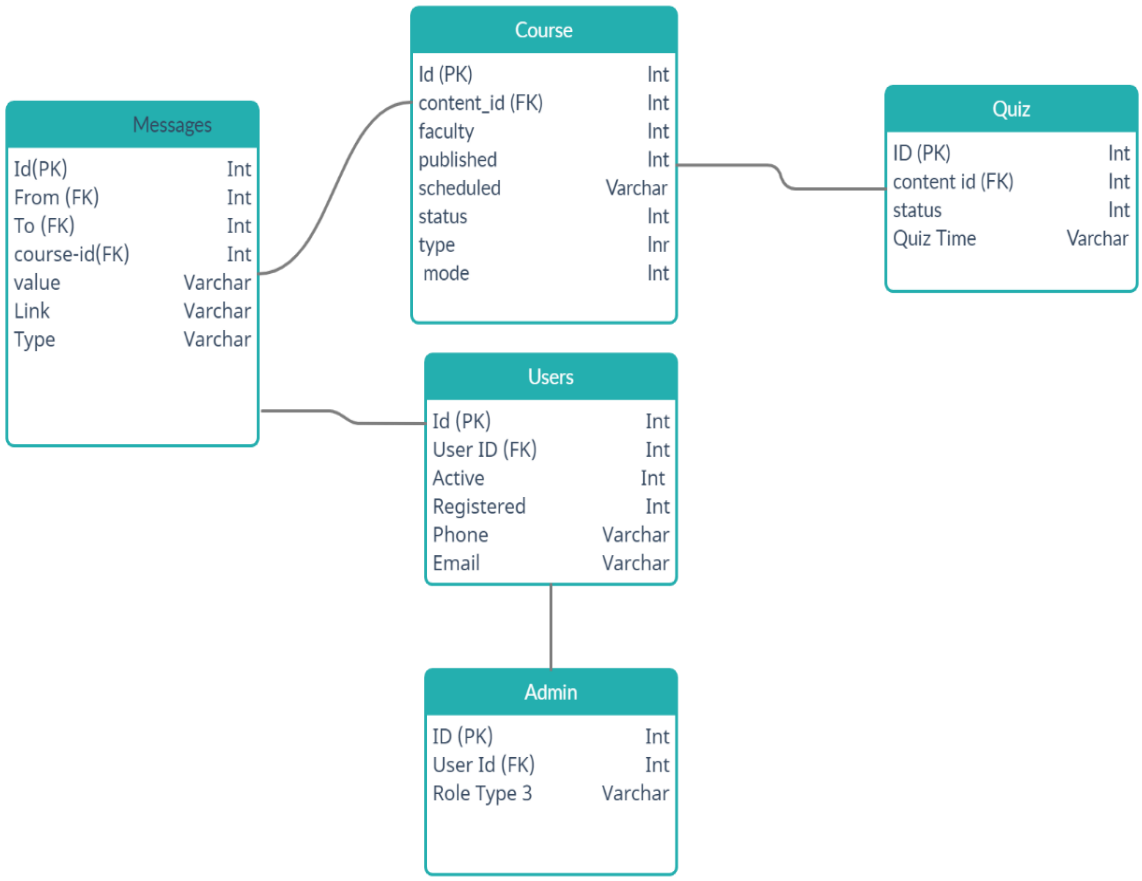

Fig 6.2.1 Database Schema

6.2 DATA DICTIONARY

A data dictionary contains metadata i.e., data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc.

Field Name	*Data Type*	*Field Size for display*
Id	int	4 Bytes
User-Id	Int	4 Bytes
User_type	Varchar	100 chars
Email	Varchar	50 chars
Password	Varchar	100 chars
Phone	Varchar	12 chars
Active	Int	4 Bytes
Registered	Int	4 Bytes
Last_Login	Char	50 chars

Fig 6.3.1 User Table

Field Name	*Data Type*	*Field Size for display*
Course_Id	Int	4 Bytes
Published	Int	4 Bytes
Schedule	Varchar	50 chars
Faculty_id	Int	4 Bytes
Content _id	Int	4 Bytes
Status	Varchar	100 chars
Type	Varchar	50 chars
Mode	Varchar	50 chars

Fig 6.3.2 Course Table

Field Name	*Data Type*	*Field Size for display*
Id	Int	4 Bytes
From	Int	4 Bytes
To	Int	4 Bytes
Type	Int	4 Bytes
Valye	Varchar	100 chars
Link	Varchar	100 chars

Figure 6.3.3 of Messages Table

Field Name	*Data Type*	*Field Size for display*
Id	int	4 Bytes
UserId	int	4 Bytes
RoleType	int	4 Bytes

Figure 6.3.4 of Admin Table

Field Name	*Data Type*	*Field Size for display*
Id	Int	4 Bytes
Content ID	Int	4 Bytes
Created At	Varchar	4 Bytes
State	Int	4 Bytes
Course ID	Varchar	50 chars
Quiz Time	Varchar	50 Chars

Figure 6.3.5 of Quiz Table

Chapter 7

TESTING

TESTING

7.1 TESTING OBJECTIVE

Software testing is an activity which aims at evaluating the quality of a software product and also to improve it by identifying defects. Software testing strives to achieve its objectives but has certain limitations. However, adherence to the established objectives ensures effective testing.

Some of the significant objectives of software testing are as follows:

1. To evaluate the work products such as requirements, design, user stories, and code
2. To verify the fulfilment of all specified requirements:
3. To validate if the test object is complete and works as per the expectation of the users and the stakeholders
4. To build confidence in the quality level of the test object
5. To prevent defects in the software product:
6. To find defects in the software product
7. To reduce the level of risk of insufficient software quality:
8. To comply with contractual, legal, or regulatory requirements or standards, and to verify the test object's compliance with such requirements or standards

7.2 KARMA: THE SPECTACULAR TEST RUNNER

Karma is an engine that runs tests against code. Although it has been written for AngularJS, it's not specifically tied to it and can be used for any JavaScript application. It's highly configurable through a JSON file and the use of various plugins.

7.3 TESTING ANGULARJS CONTROLLERS

The karma testing framework also has the functionality to test Controllers end to end. This includes testing of the $scope object which is used within Controllers.
We would first need to define a controller. This controller would carry out the below-mentioned steps

1. Create an ID variable and assign the value 5 to it.
2. Assign the ID variable to the $scope object.

Our test will test the existence of this controller and also test to see if the ID variable of the $scope object is set to 5.

First, we need to ensure the following prerequisite is in place

1. Install the Angularjs-mocks library via npm. This can be done by executing the below line in the command prompt npm install Angular JS-mocks
2. Next is to modify the karma.conf.js file to ensure the right files are included for the test. The below segment just shows the files part of the karma.conf.js which needs to be modified
   files: ['lib/AngularJS.js','lib/AngularJS-mocks.js','lib/index.js','test/*.js']
   - The 'files' parameter basically tells Karma all the files that are required in the running of the tests.
   - The AngularJS.js and AngularJS-mocks.js file are required to run AngularJS unit tests
   - The index.js file is going to contain our code for the controller
   - The test folder is going to contain all our AngularJS tests

Below is our Angular.JS code which will be stored as a file Index.js in the test folder of our application.

The below code just does the following things

1. Create an Angular JS module called sampleApp
2. Create a controller called AngularJSController
3. Create a variable called ID, give it a value of 5 and assign it to the $scope object

```
var sampleApp = AngularJS.module('sampleApp',[]);

sampleApp.controller('AngularJSController', function($scope) {

   $scope.ID =5;

 });
```

Once the above code is executed successfully, the next step would be to create a Test Case to ensure the code has been written and executed properly.

The code for our test will be as shown below.

The code will be in a separate file called ControllerTest.js, which will be placed in the test folder. The below code just does the following key things

1. **beforeEach function** – This function is used to load our AngularJS.JS module called 'sampleApp' before the test run. Note that this is the name of the module in an index.js file.
2. The **$controller** object is created as a mockup object for the controller "Angular JSController" which is defined in our index.js file. In any sort of Unit Testing, a mock object represents a dummy object which will actually be used for the testing. This mock object will actually simulate the behavior of our controller.
3. **beforeEach(inject(function(_$controller_)** – This is used to inject the mock object in our test so that it behaves like the actual controller.
4. **var $scope** = {}; This is a mock object being created for the $scope object.

5. **var controller** = $controller('AngularJSController', { $scope: $scope }); - Here we are checking for the existence of a controller named 'Angular.JSController'. In here we are also assigning all variables from our $scope object in our controller in the Index.js file to the $scope object in our test file
6. Finally, we are comparing the $scope.ID to 5

```
describe('AngularJSController', function() {
    beforeEach(module('sampleApp'));
    var $controller;
    beforeEach(inject(function(_$controller_){
            $controller = _$controller_;
    }));
    describe('$scope.ID', function() {
```

```
        it('Check the scope object', function() {
            var $scope = {};
           var controller = $controller('AngularJSController', { $scope: $scope });
            expect($scope.ID).toEqual(5);
        });
    });
});
```

7.4 END TO END TESTING ANGULARJS JS APPLICATIONS

The karma testing framework along with a framework called Protractor has the functionality of testing a web application end to end.So it's not only testing of directives and controllers, but also testing of anything else which may appear on an HTML page.Let's look at an example of how we can achieve this.In our example below, we are going to have an AngularJS application which creates a data table using the ng-repeat directive.

1. We are first creating a variable called "tutorial" and assigning it some key-value pairs in one step. Each key-value pair will be used as data when displaying the table. The tutorial variable is then assigned to the scope object so that it can be accessed from our view.
2. For each row of data in the table, we are using the ng-repeat directive. This directive goes through each key-value pair in the tutorial scope object by using the variable ptutor.
3. Finally, we are using the <td> tag along with the key value pairs (ptutor.Name and ptutor.Description) to display the table data.

Once the above code is executed successfully, the next step would be to create a test case to ensure the code has been written and executed properly. The code for our test will be as shown belowOur test is actually going to test the ng-repeat directive and ensure that it contains 3 rows of data as it should from the above example.

7.5 TEST CASES

7.5.1 Login Test Case

Steps	Test Steps	Test Data	Expected Results	Actual Result	Status
1.	Email and password is Correct	Email and Password	Login Successfully	Login Successfully	Pass
2.	Email is Incorrect	Email	Show Toast Message	Show Toast Message	Pass
3.	Password is Incorrect	Password	Show Toast Message	Show Toast Message	Pass
4.	Attempt Login with Wrong Password Many times	Email and Pass	Block User for some time	User Blocked	Pass

Figure 7.5.1 for Login Test Case.

7.5.2 Chat Test Case

Steps	Test Steps	Test Data	Expected Results	Actual Result	Status
1.	Send a Empty Message	Message	Should not sent to server	Message is not displayed ion chat	Pass
2.	Send an Image	Image	Image Sent	We can see Image in chat box	Pass
3.	Send an Message with Image	Password and Message	Image and Message sent	We can see both images and messages in chat box	Pass

Figure 7.5.2 for Chat Test Case.

Chapter 8

SNAPSHOTS OF GUI

SNAPSHOTS OF GUI

8.1 BASIC FLOW

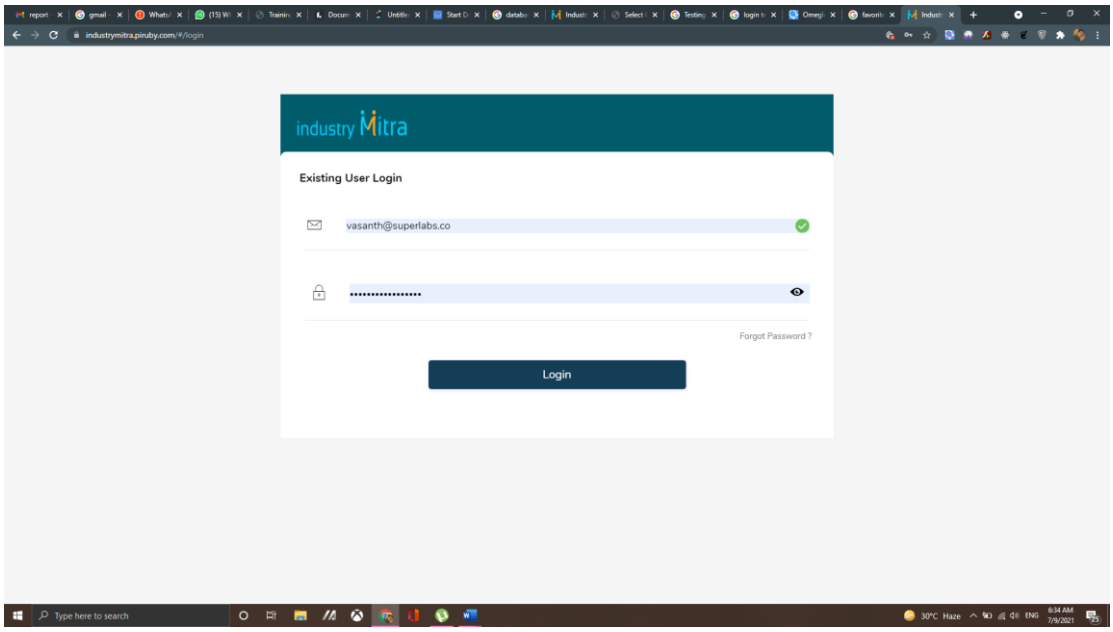

Fig 8.1.1 Login Screen

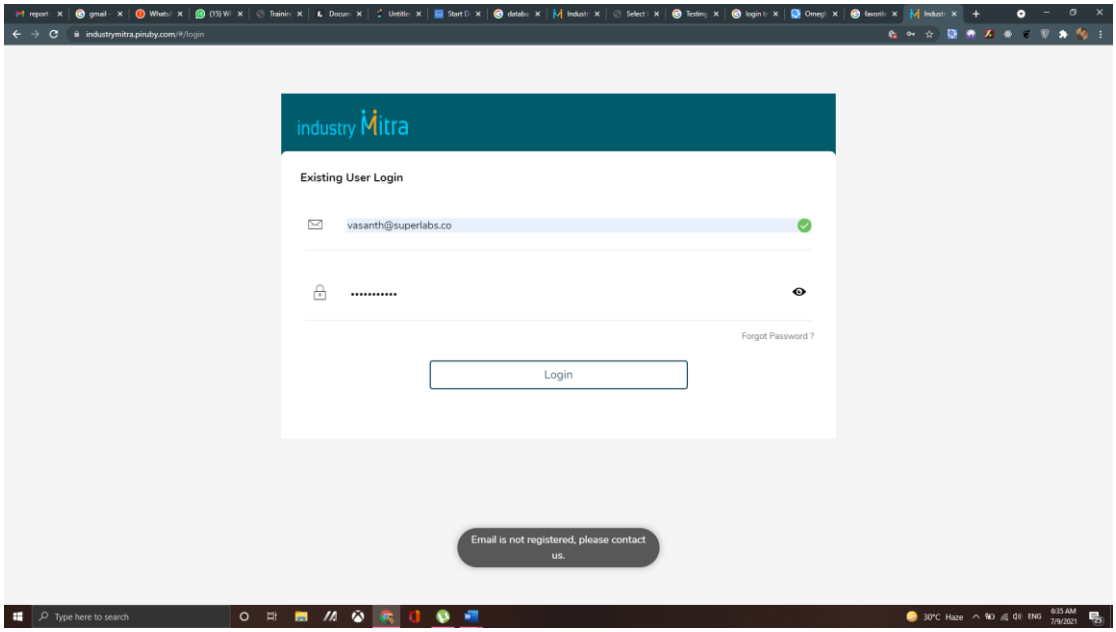

Fig 8.1.2 Wrong Password Entered

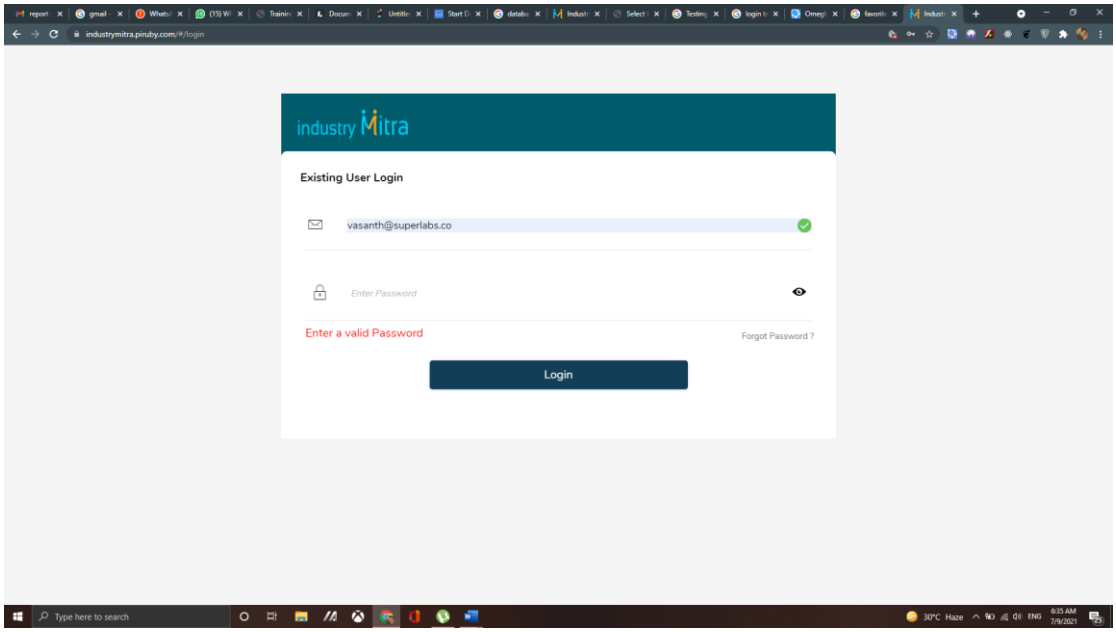

Fig 8.1.3 Password field Empty

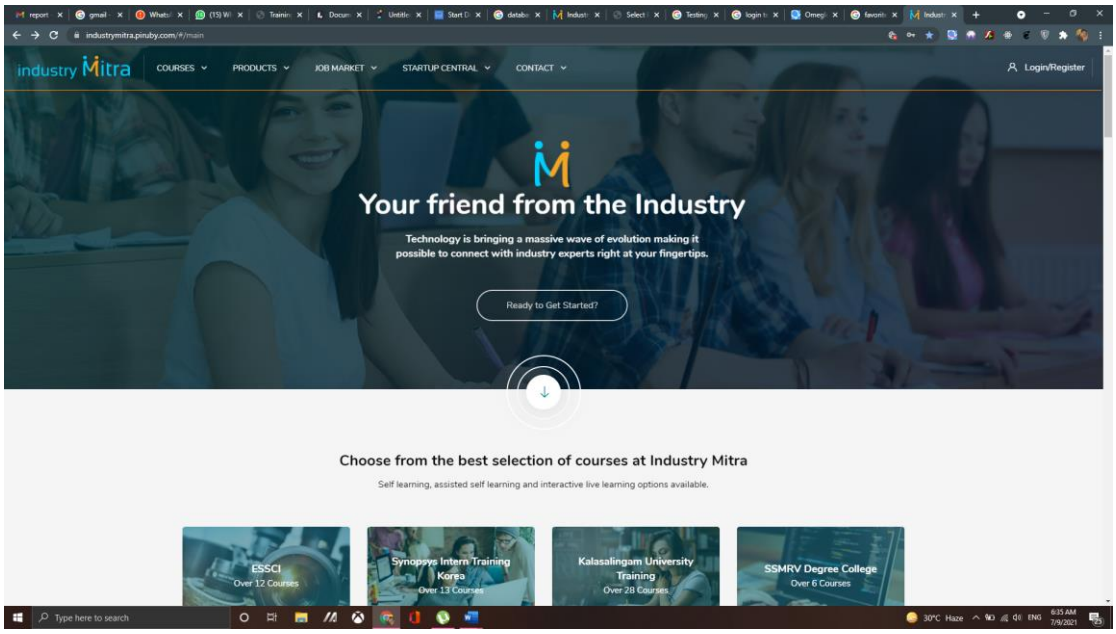

Fig 8.1.4 Home Page

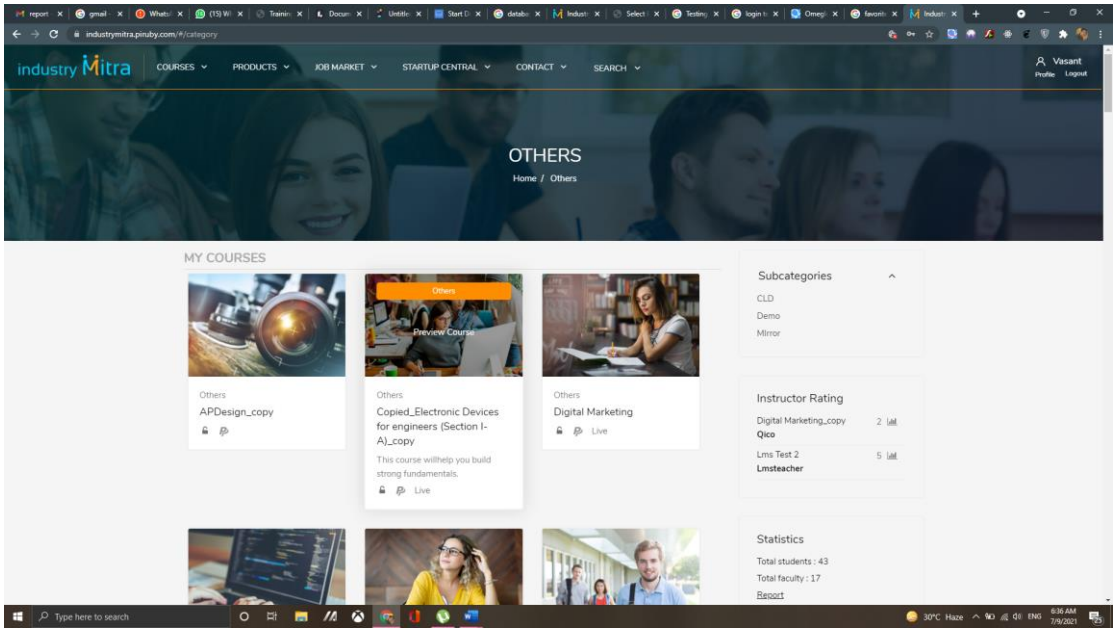

Fig 8.1.5 Category Page

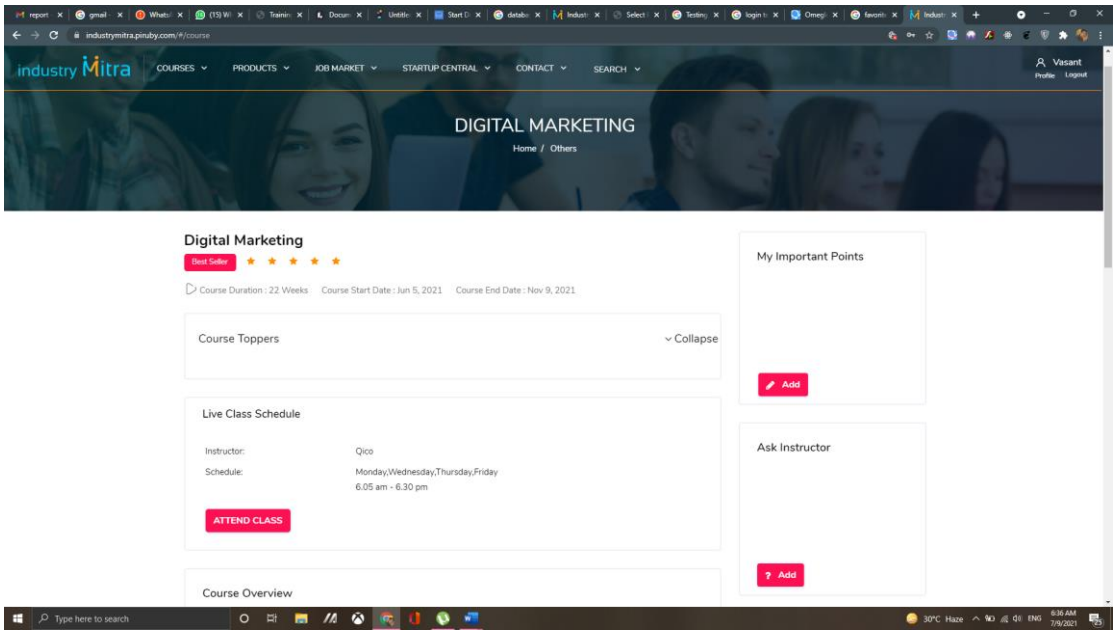

Fig 8.1.6 Course Page

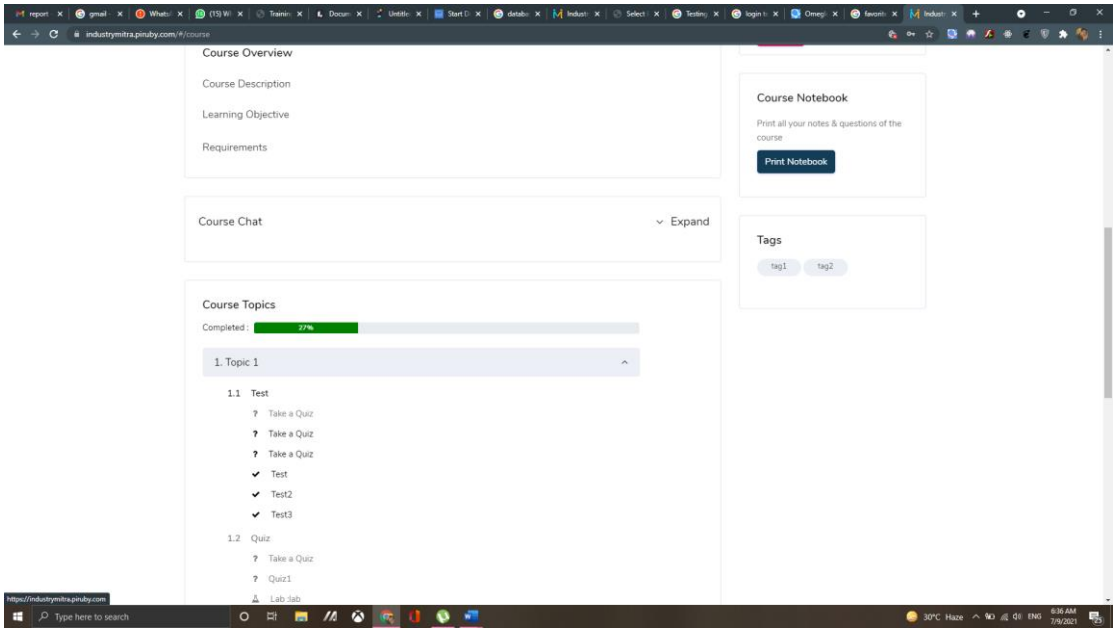

Fig 8.1.7 Topics and Sub topics

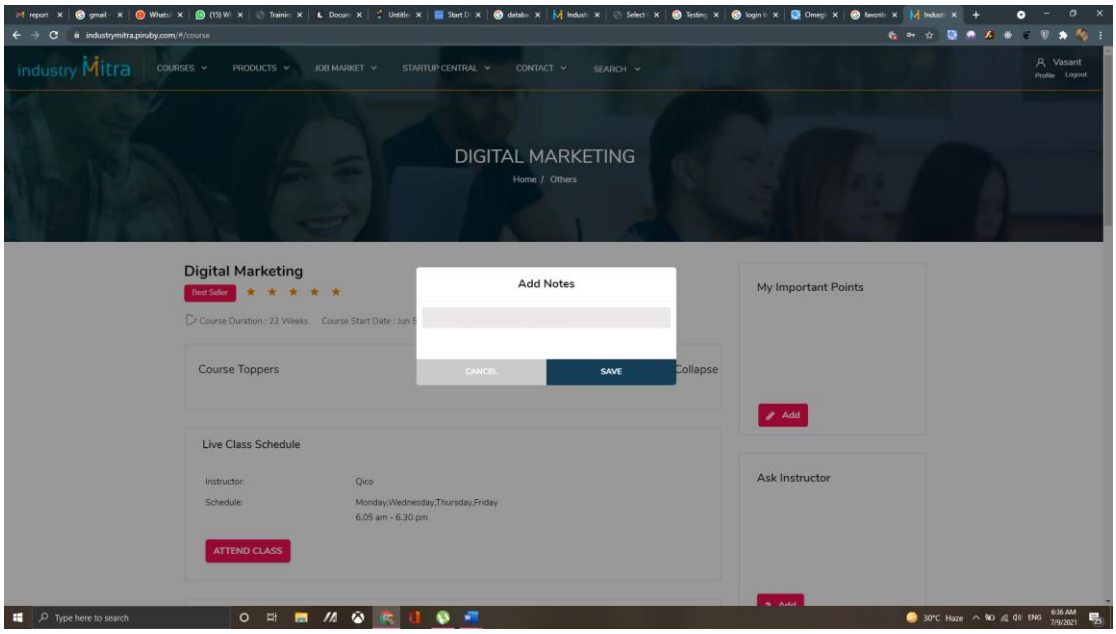

Fig 8.1.8 Add Notes

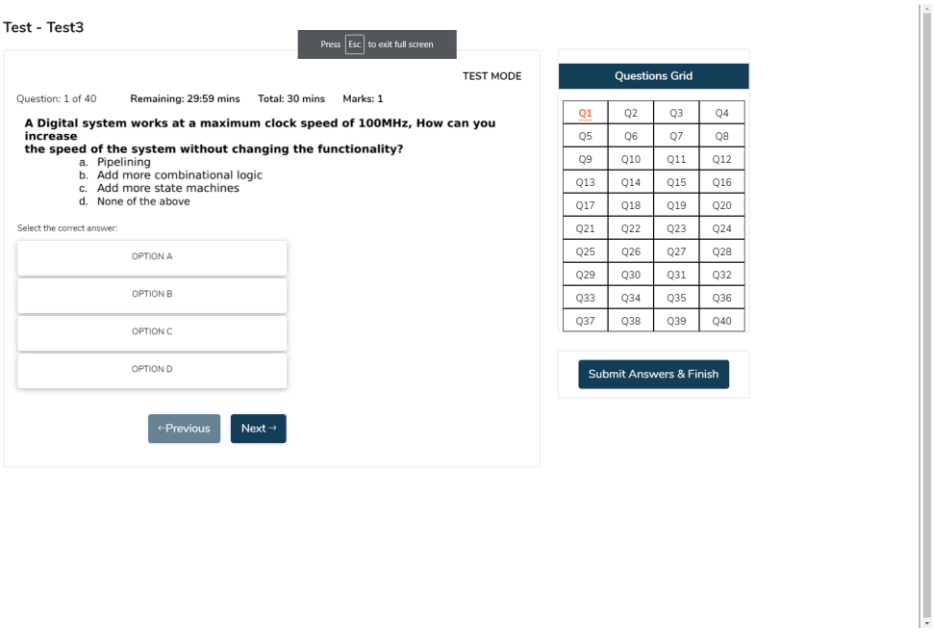

Fig 8.1.9 Take Quiz (Student)

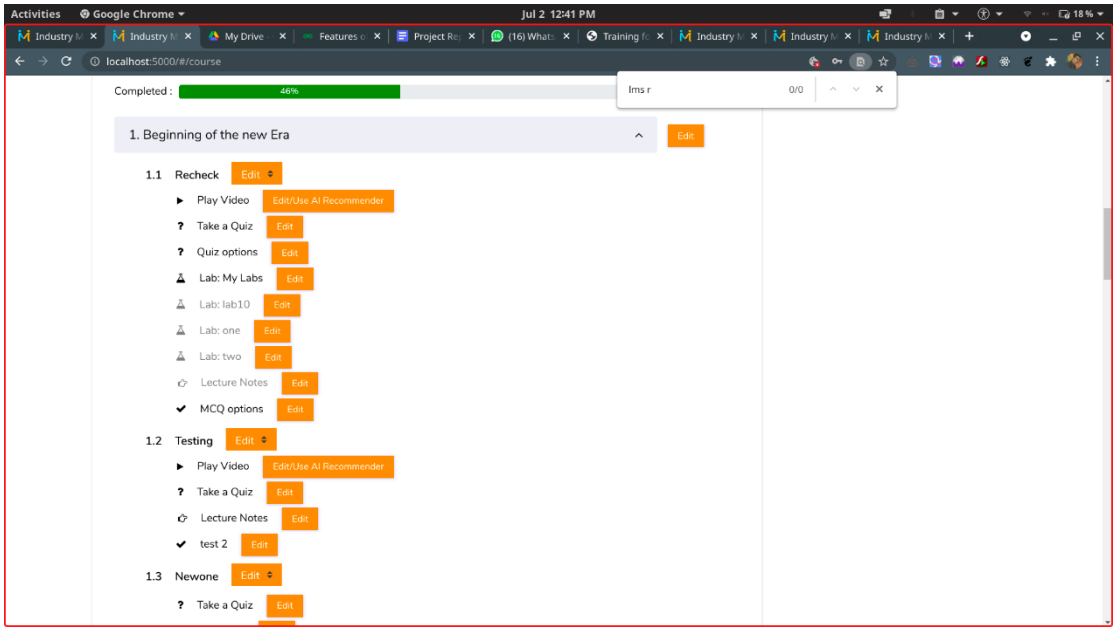

Fig 8.1.10 Topics and Subtopics (Author Mode)

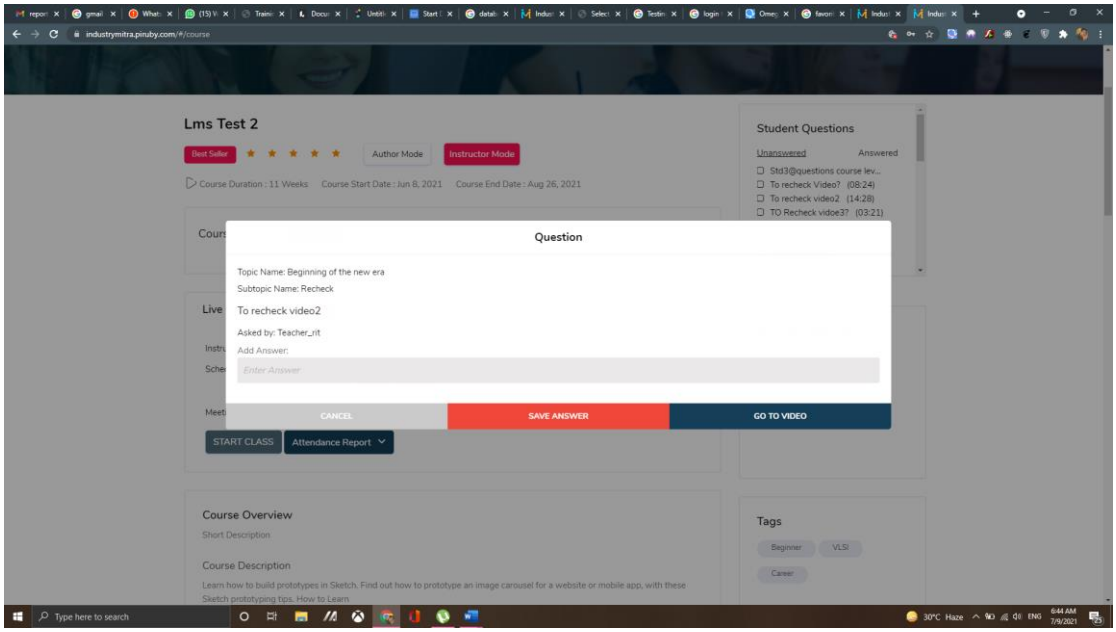

Fig 8.1.11 Answer Student Questions

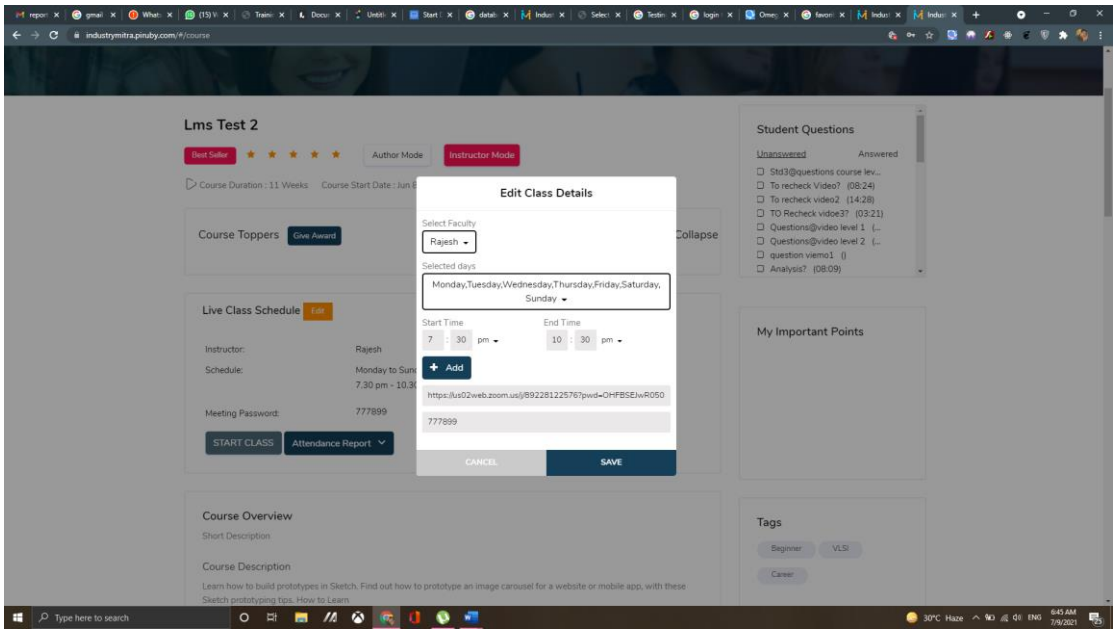

Fig 8.1.11 Edit Class Schedule (Author Mode)

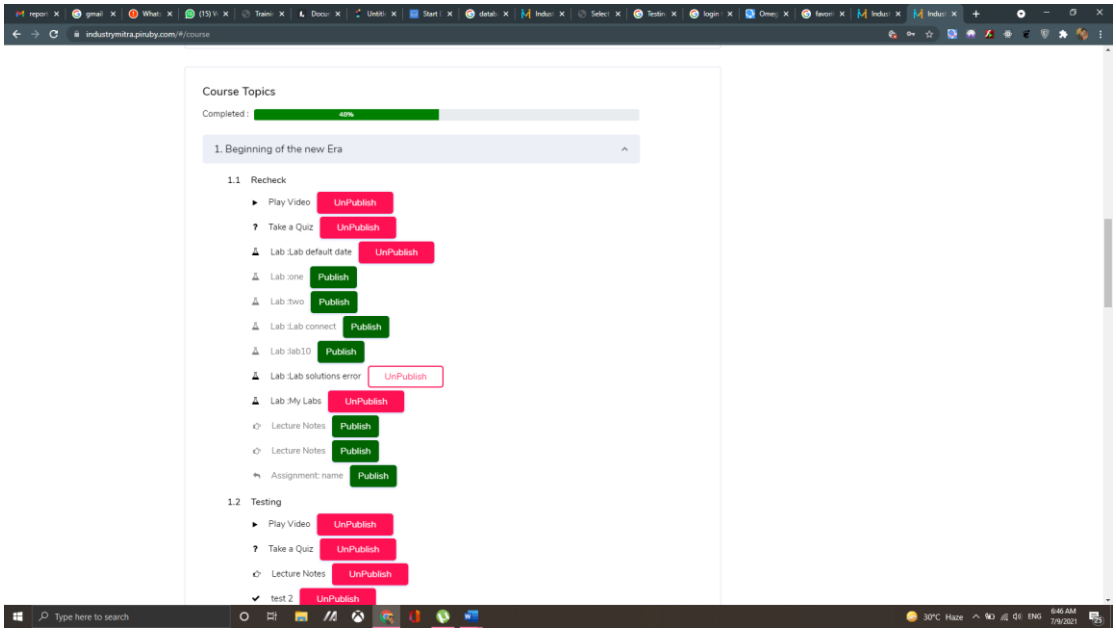

Fig 8.1.12 Topics and Subtopics (Instructor Mode)

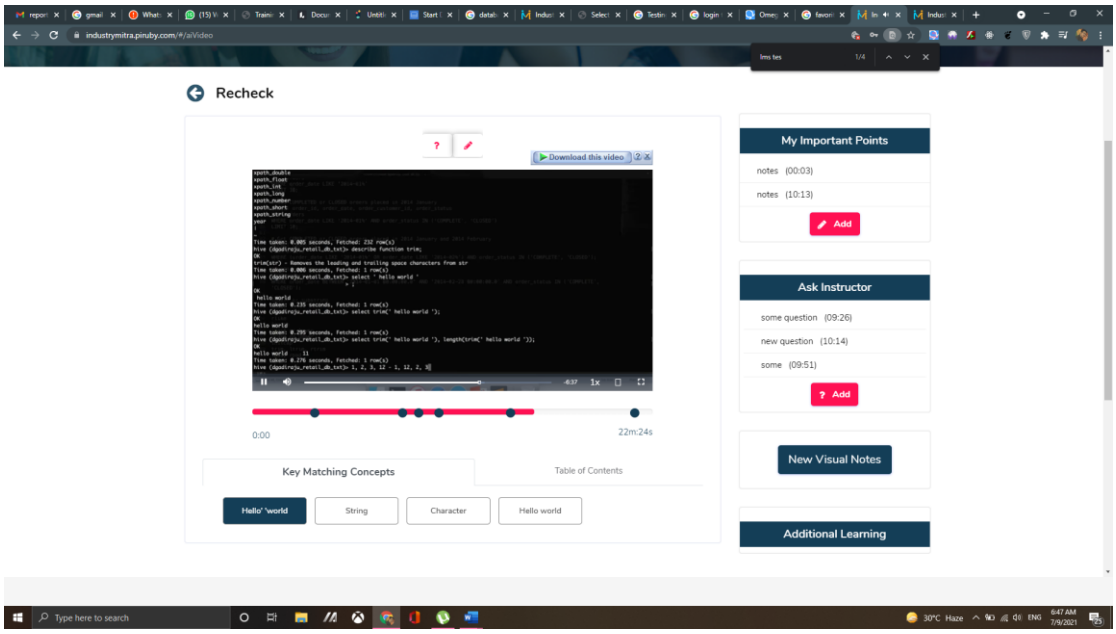

Fig 8.1.13 Play Video

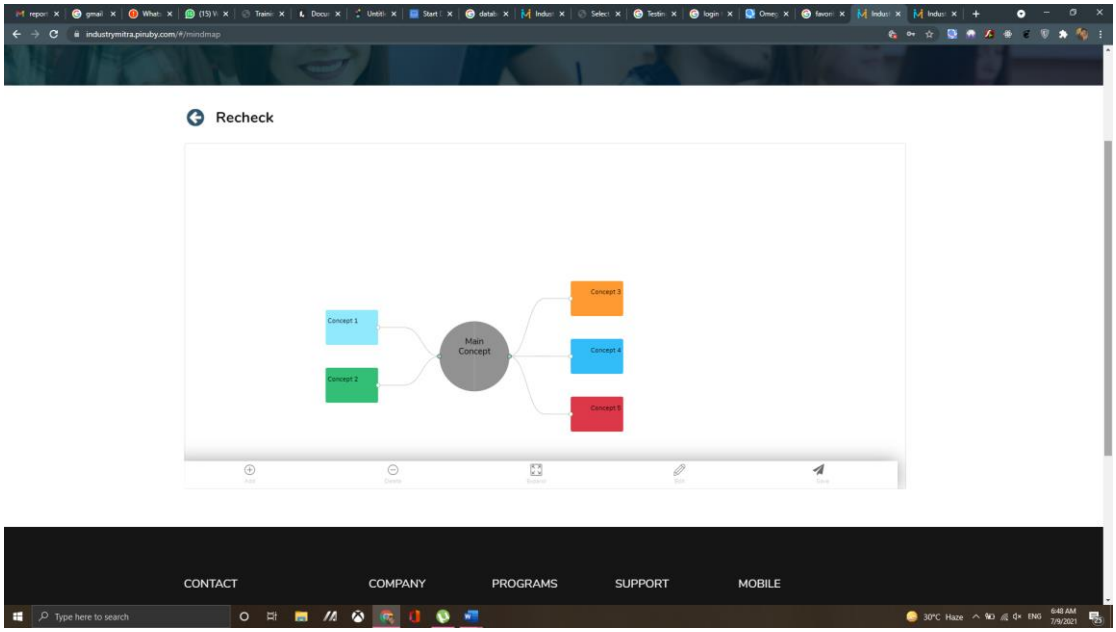

Fig 8.1.14 Visual Notes

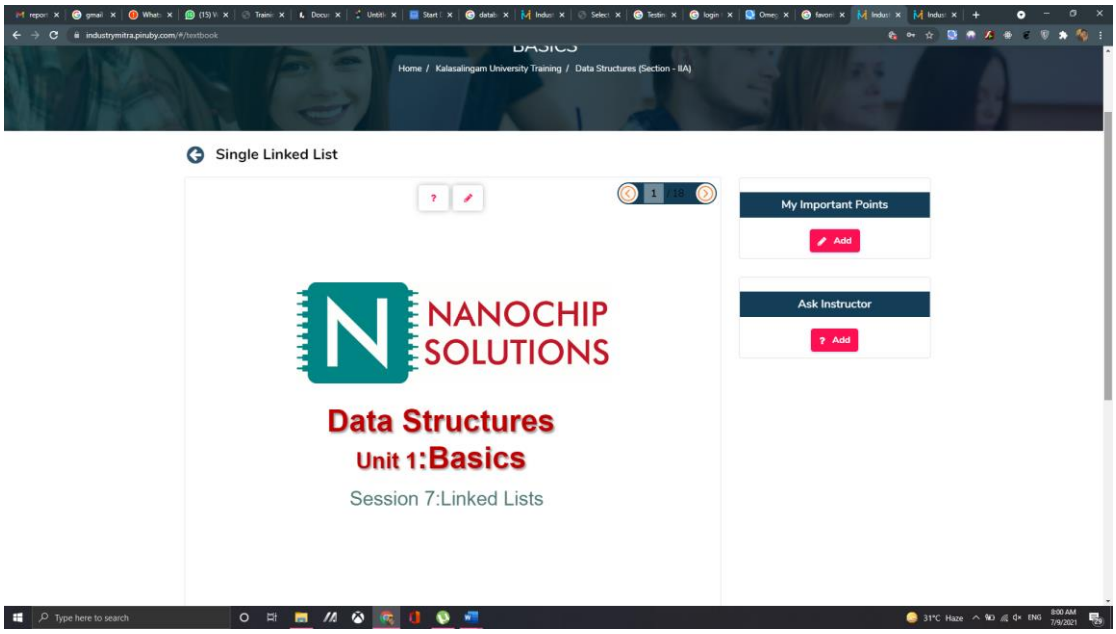

Fig 8.1.15 Lecture Notes

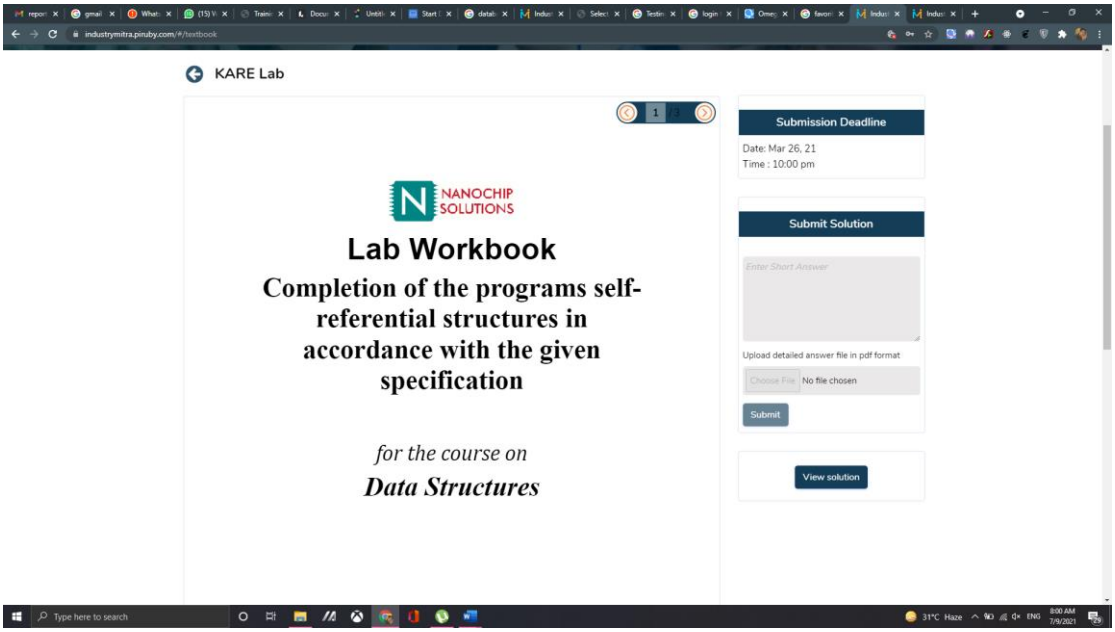
KARE Lab
NANOCHIP SOLUTIONS
Lab Workbook
Completion of the programs self-referential structures in accordance with the given specification
for the course on
Data Structures
Submission Deadline
Date: Mar 26, 21
Time : 10:00 pm
Submit Solution
Enter Short Answer
Upload detailed answer file in pdf format
Choose File
No file chosen
Submit
View solution

Fig 8.1.16 Lab

8.2 MOBILE BROWSER

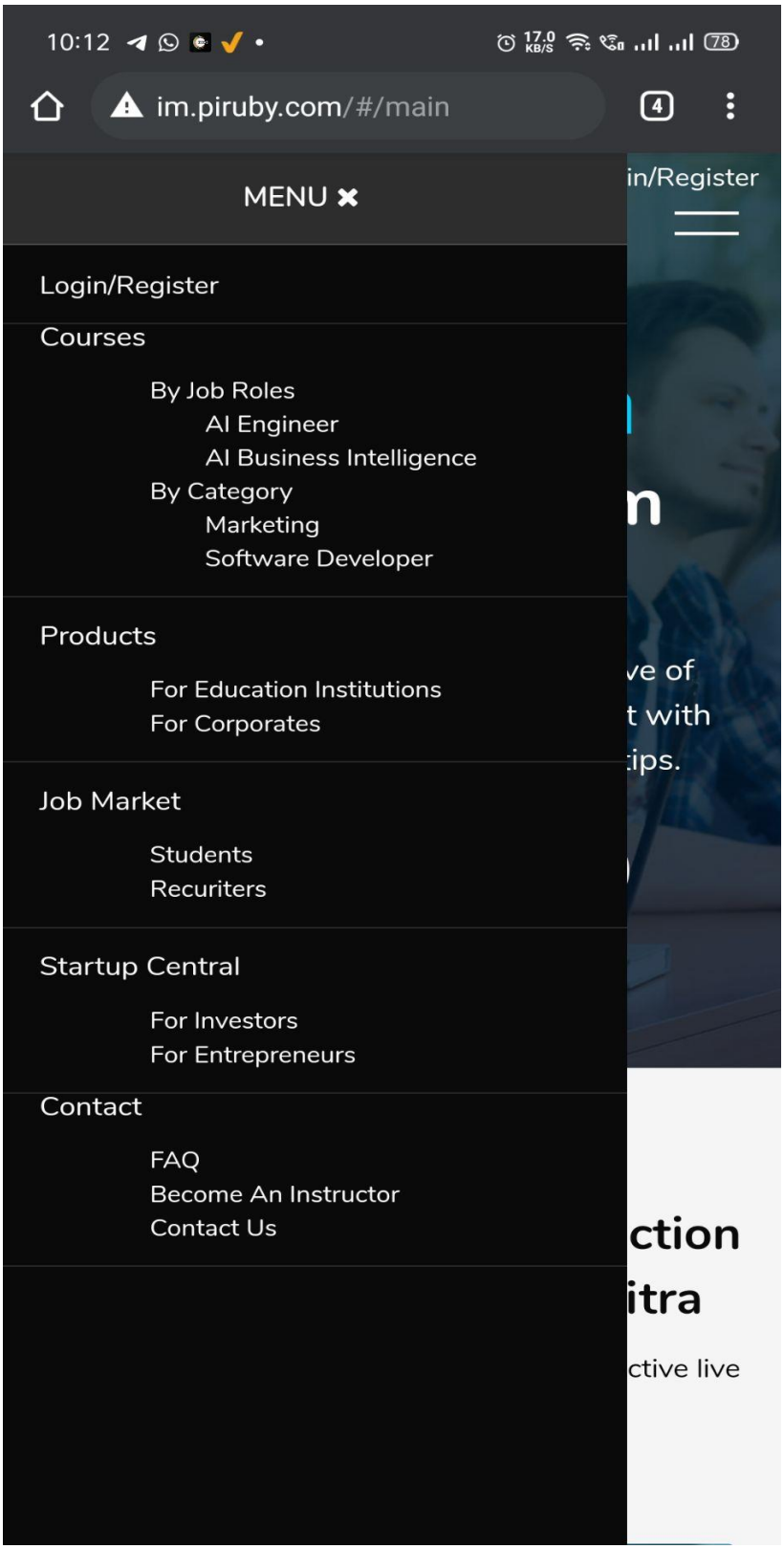

Fig 8.2.1 Hamburger Menu

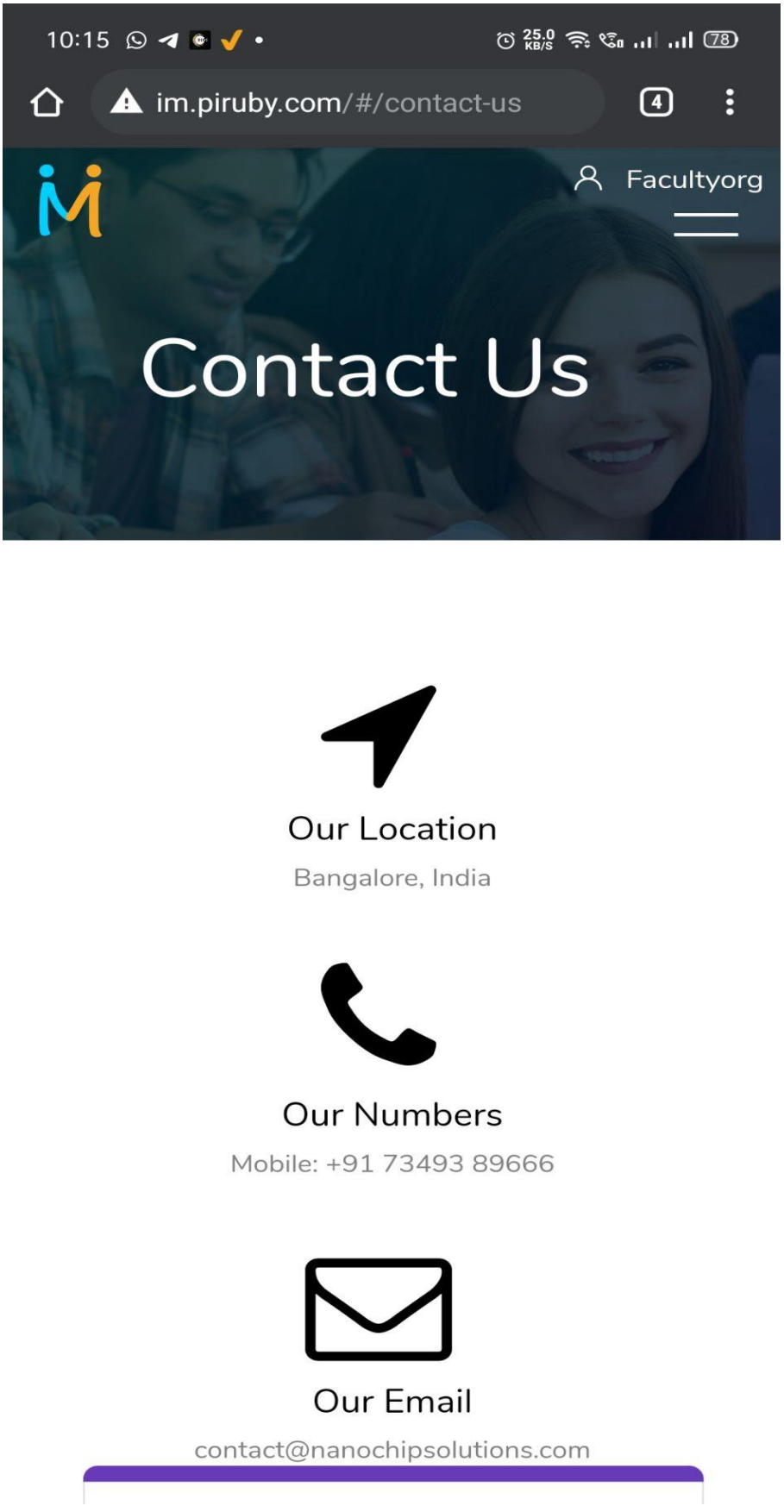

Fig 8.2.2 Contact Us Page

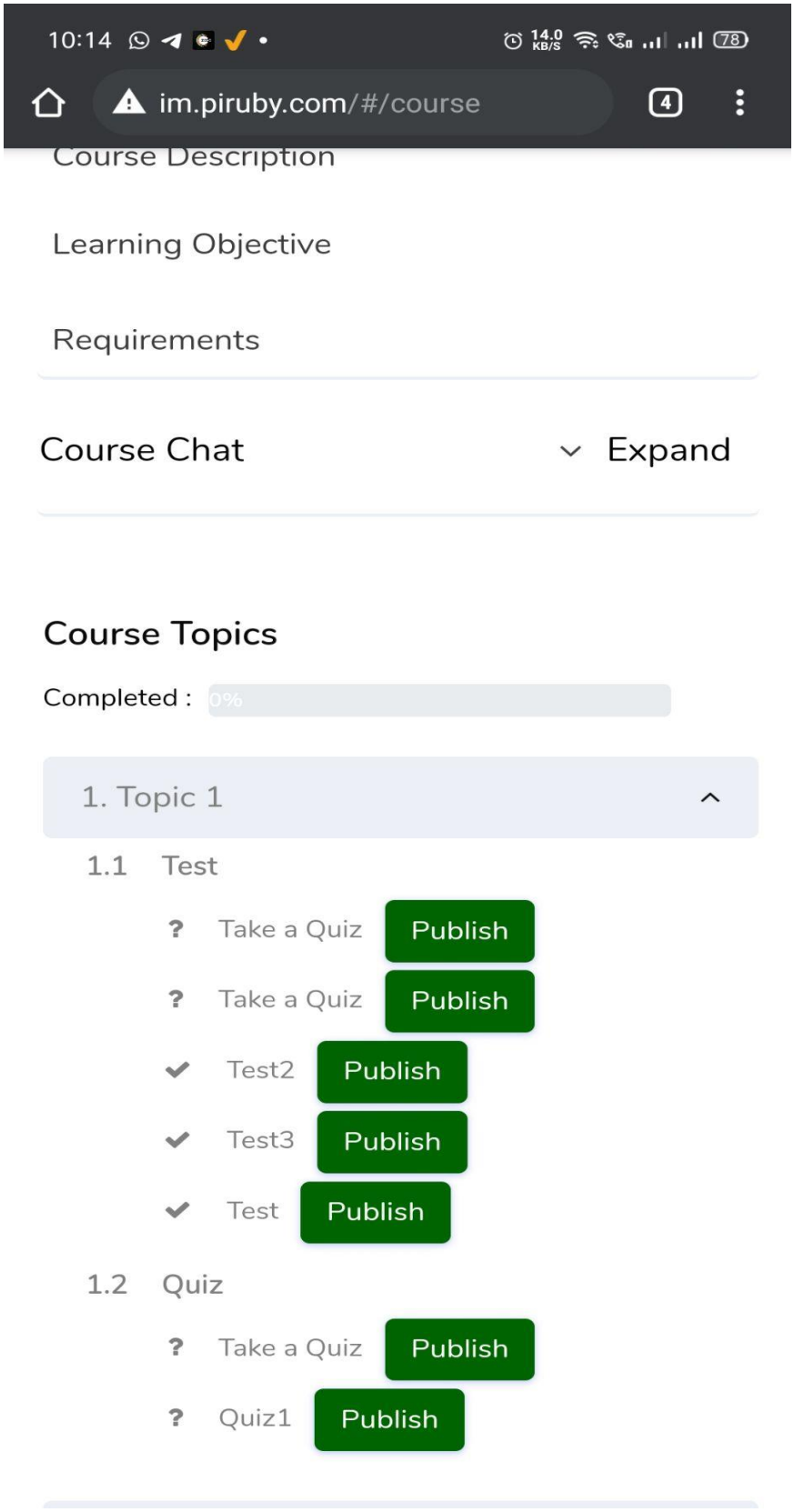

Fig 8.2.3 Instructor Mode

8.3 QUIZ MODULE

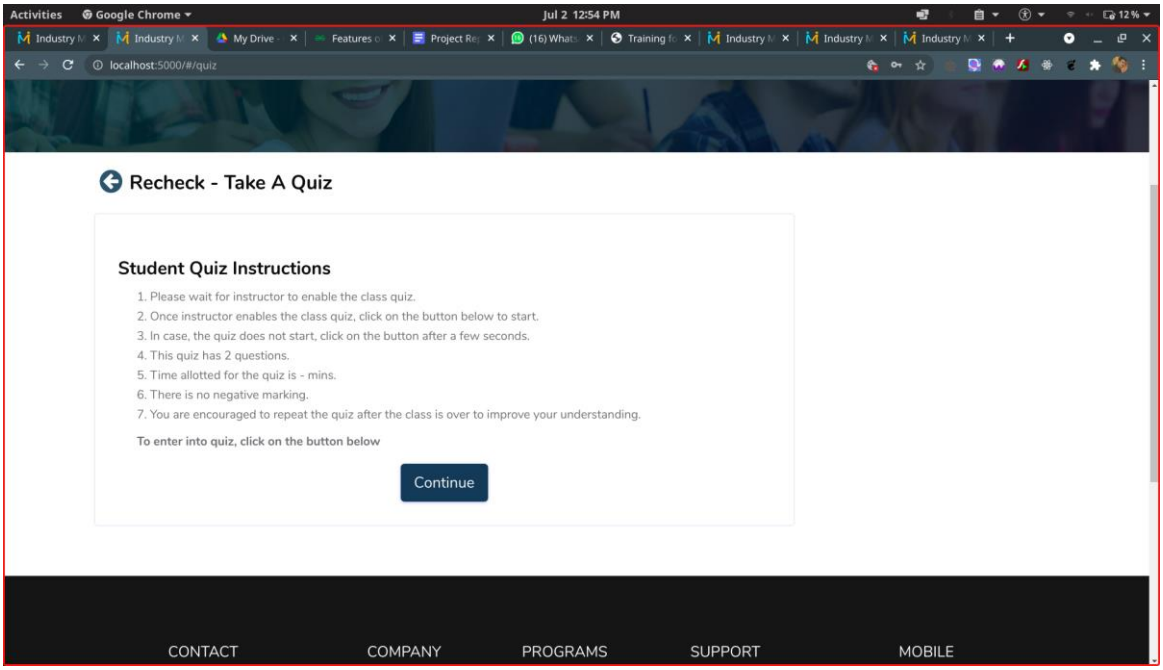

Fig 8.3.1 Start Quiz

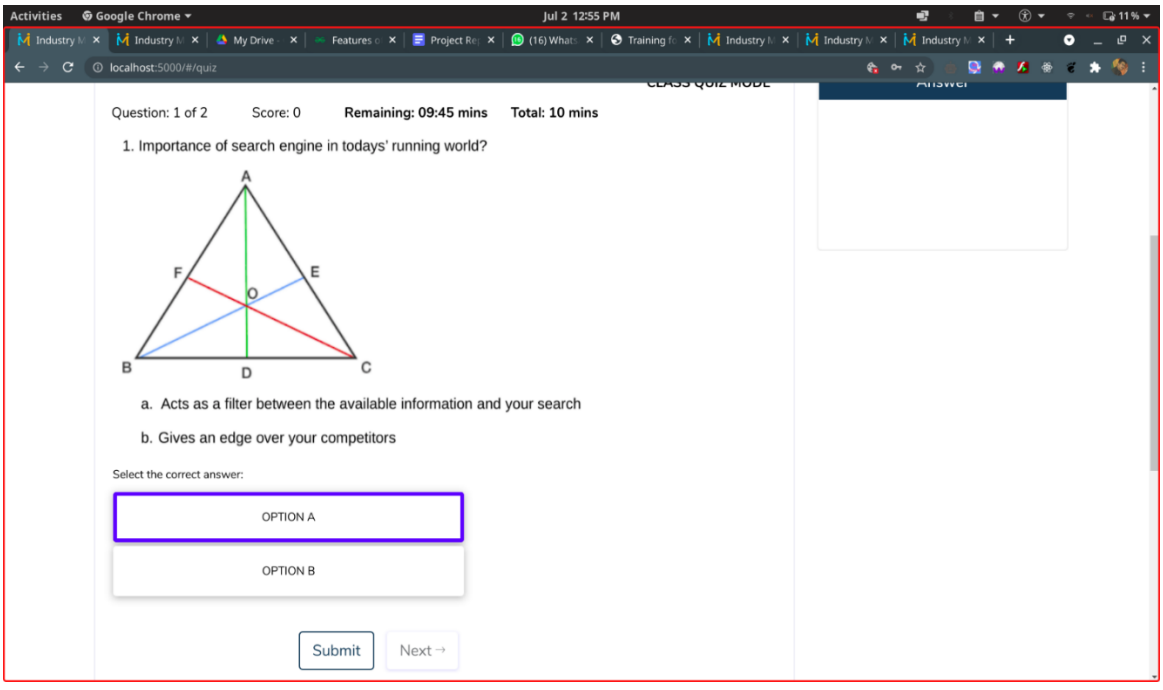

Fig 8.3.2 Quiz (Student View)

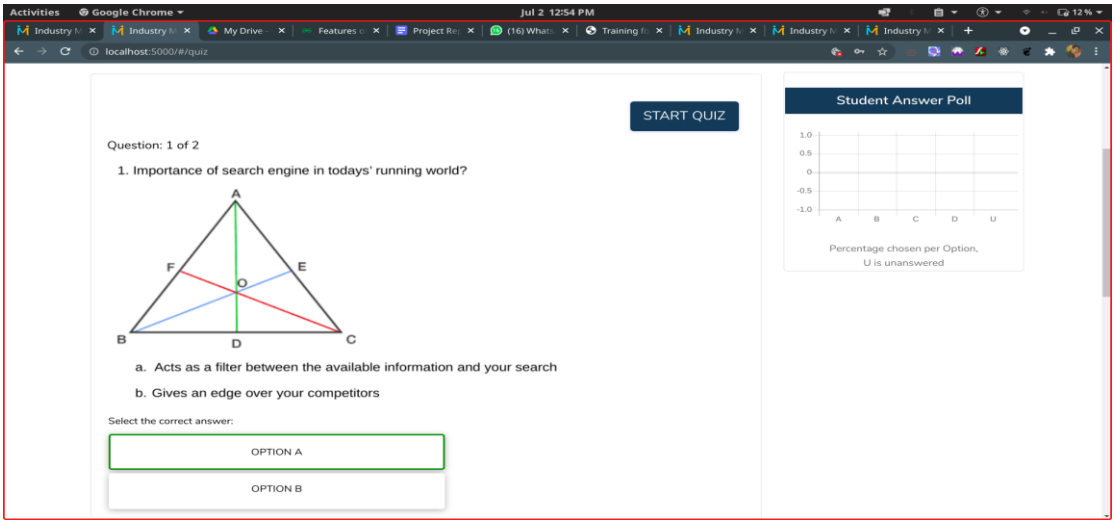

Fig 8.3.3 Quiz (Faculty View)

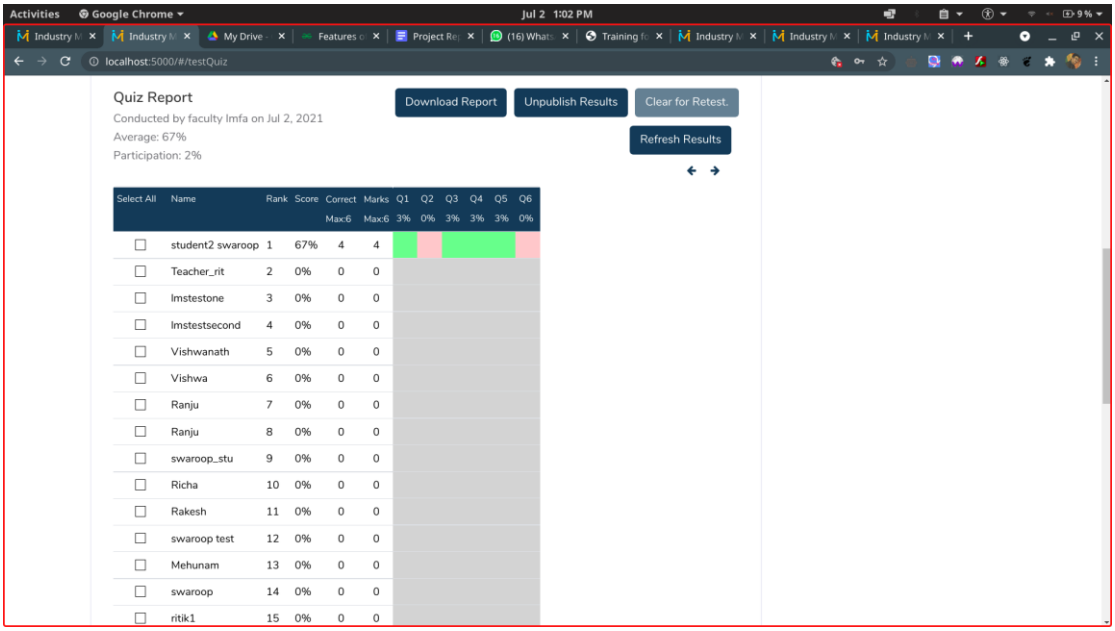

Fig 8.3.3 Check Student Answers

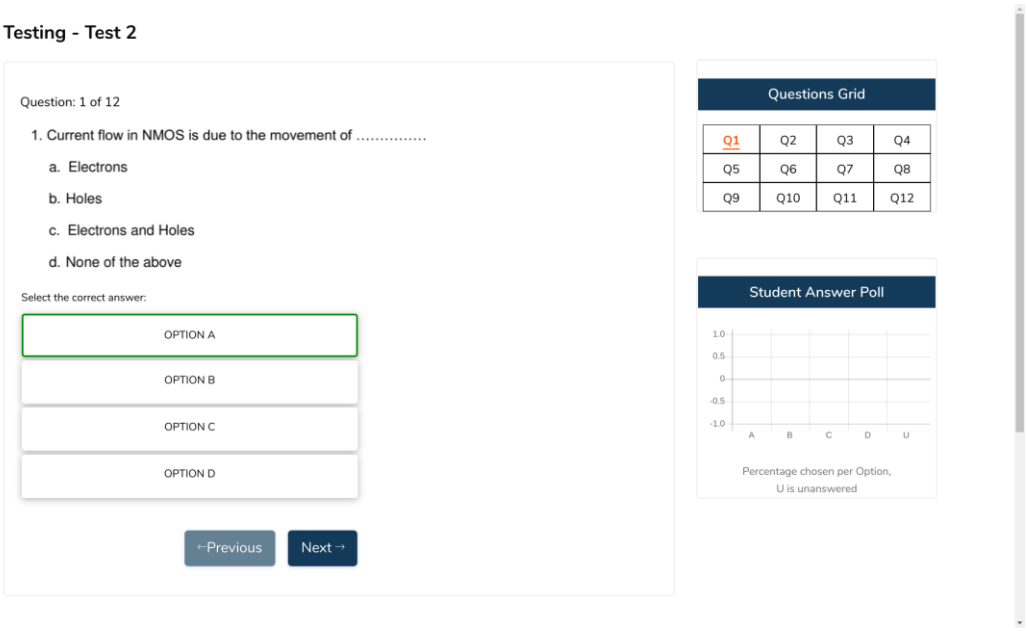

Fig 8.3.4 Mcq Mode (Faculty)

8.4 CHAT MODULE

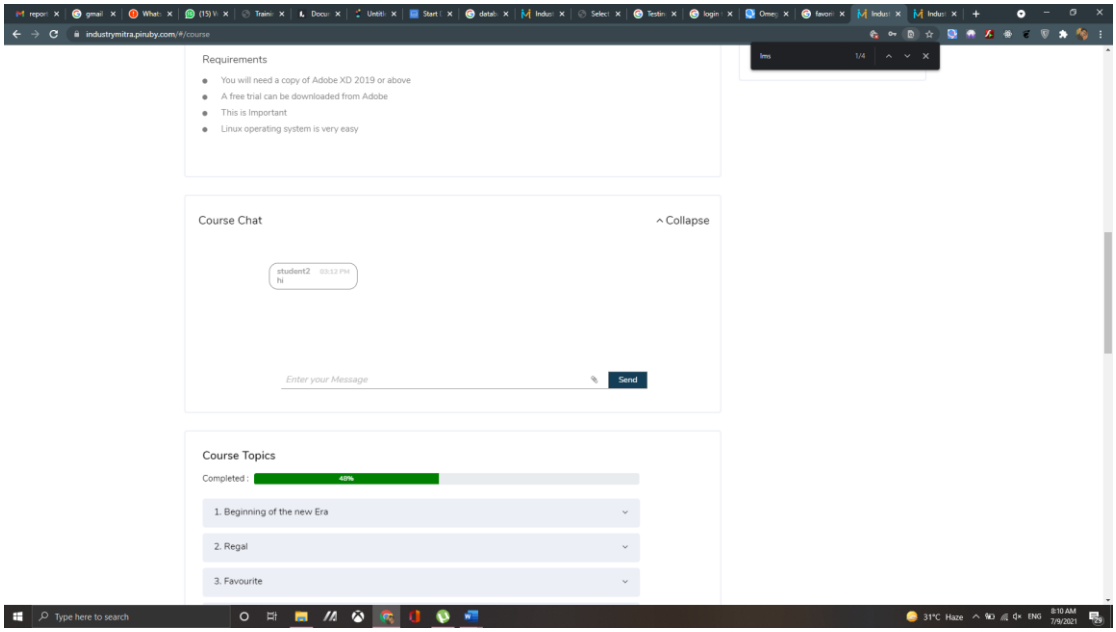

Fig 8.4.1 Chat Box

8.5 ADMIN MODULE

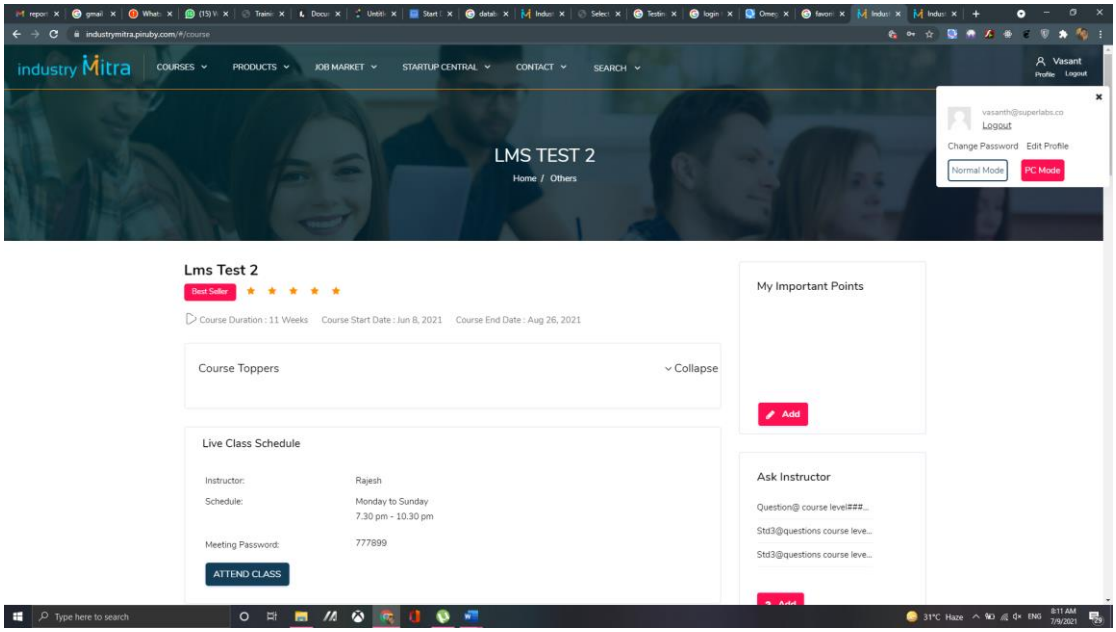

Fig 8.5.1 Toggle Admin Mode

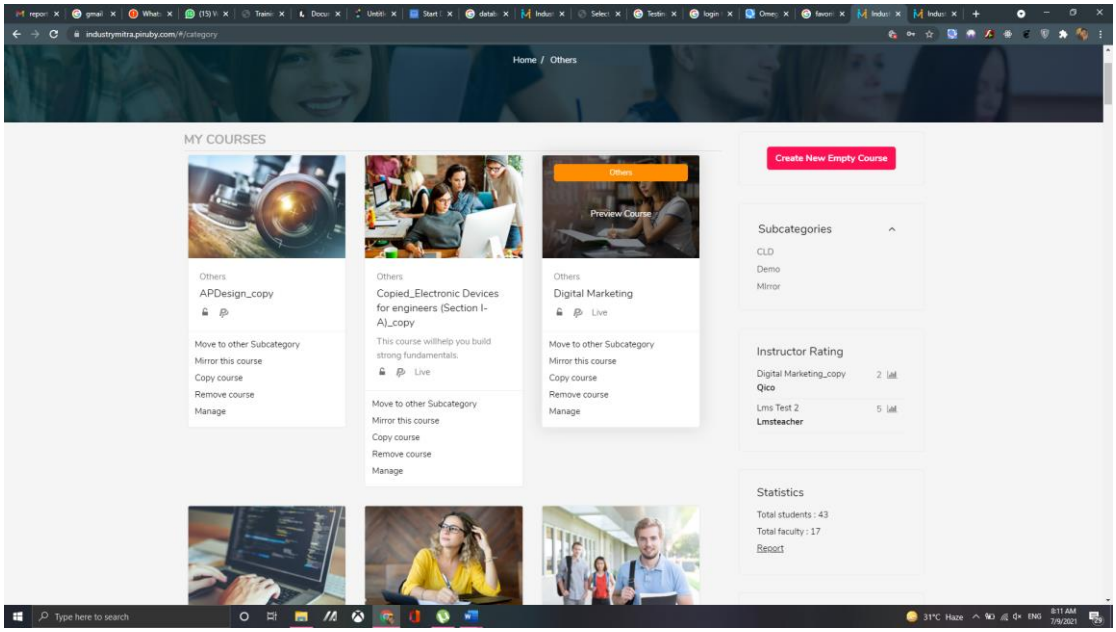

Fig 8.5.2 Manage Course

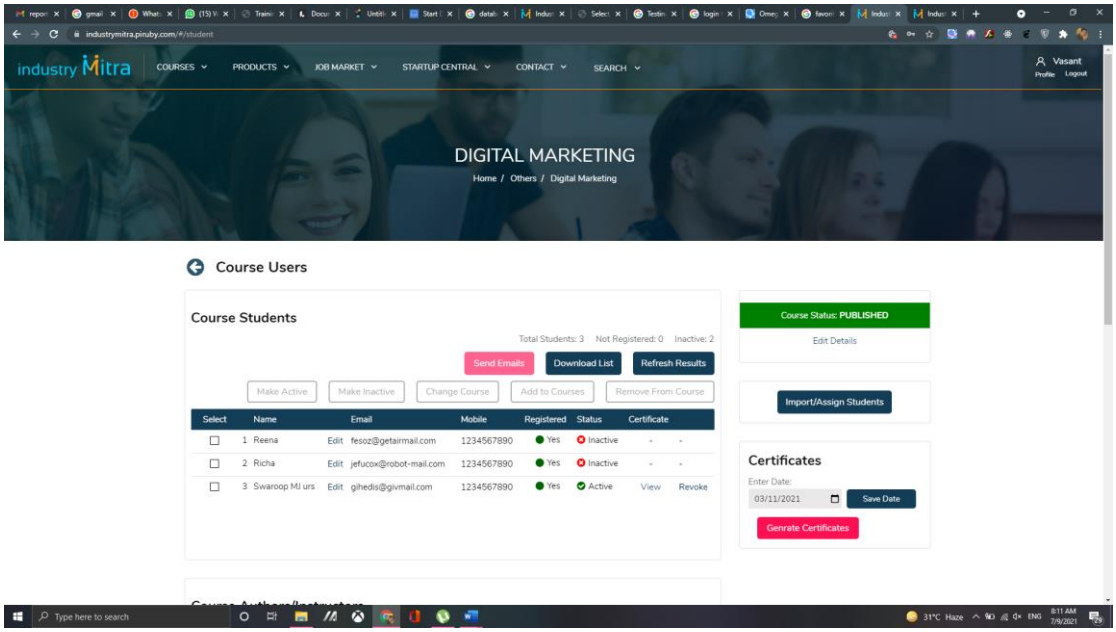

Fig 8.5.3 Manage Users

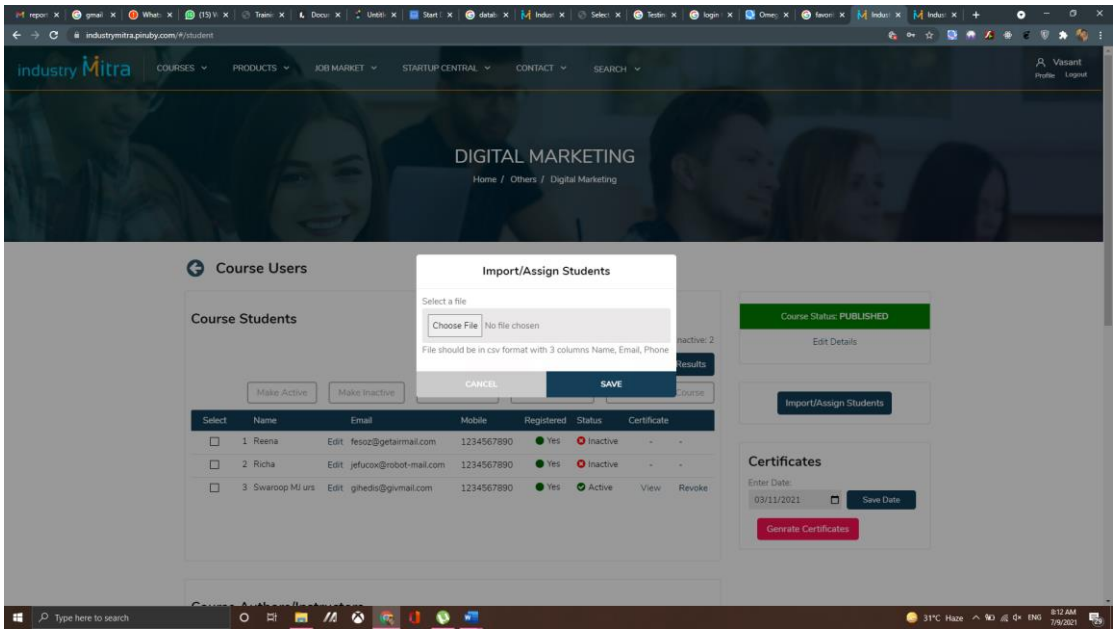

Fig 8.5.4 Import Users

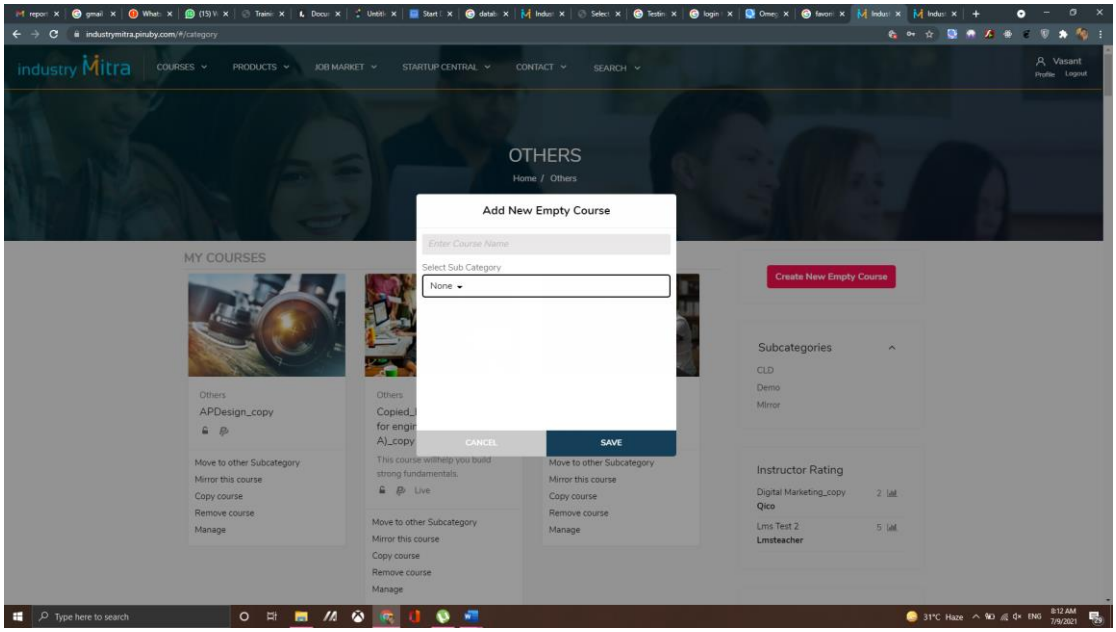

Fig 8.5.5 Add New Course

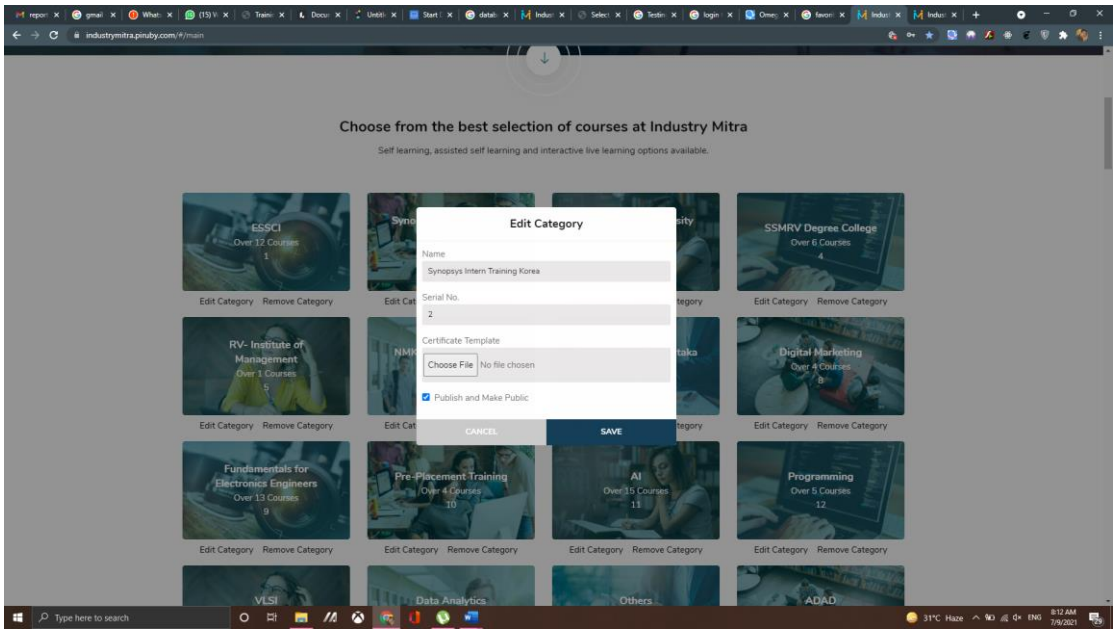

Fig 8.5.6 Add New Category

8.6 ATTENDANCE MODULE

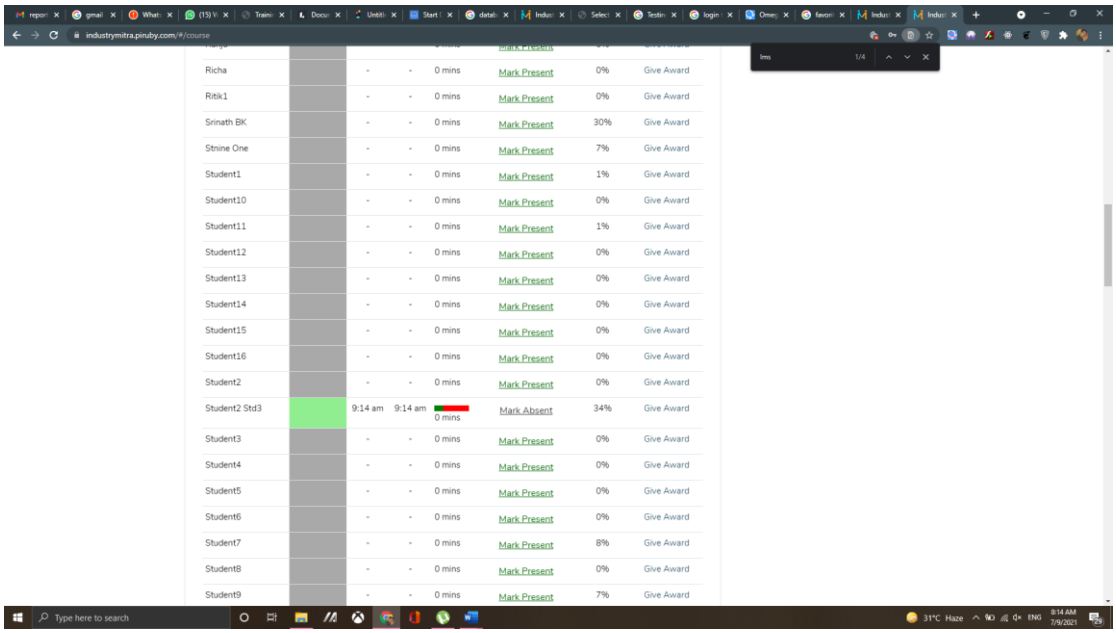

Fig 8.6.1 Check Attendance of Student

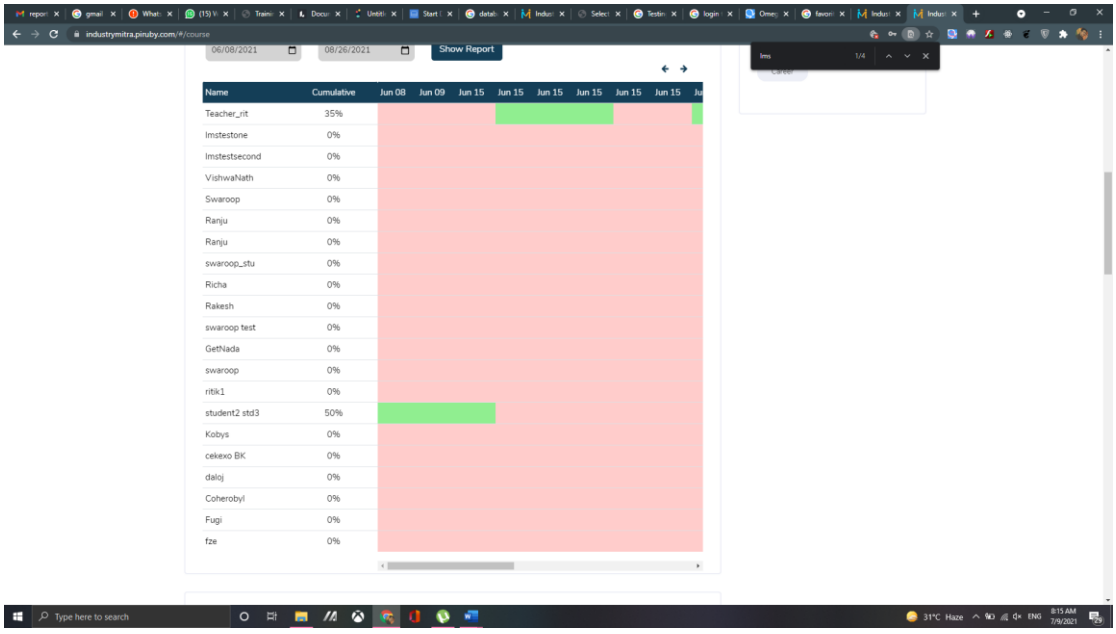

Fig 8.6.2 Check Cumulative Attendance

8.7 RATING MODULE

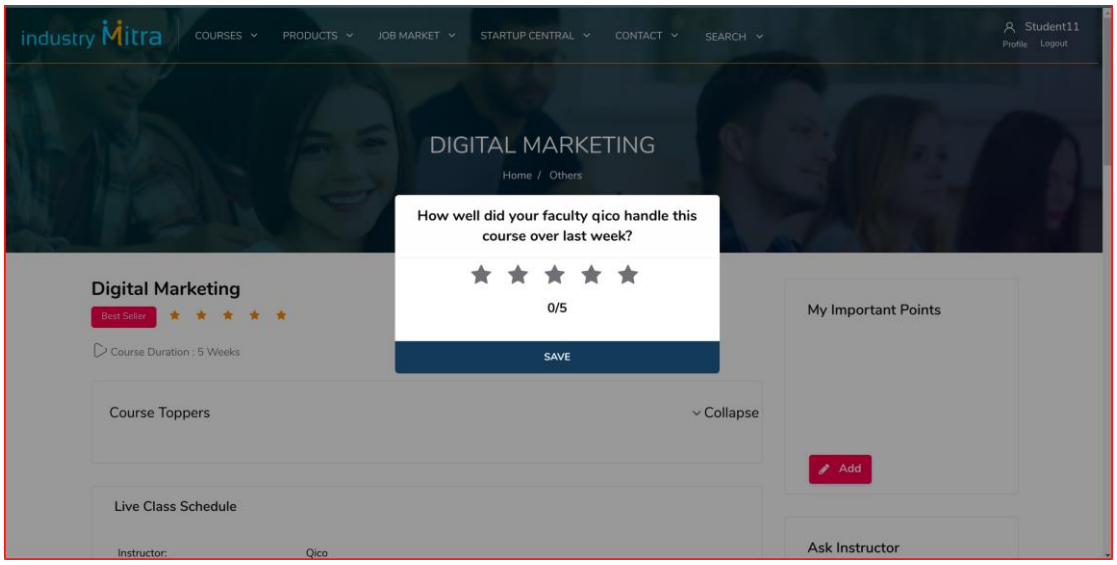

Fig 8.7.1 Give rating to faculty

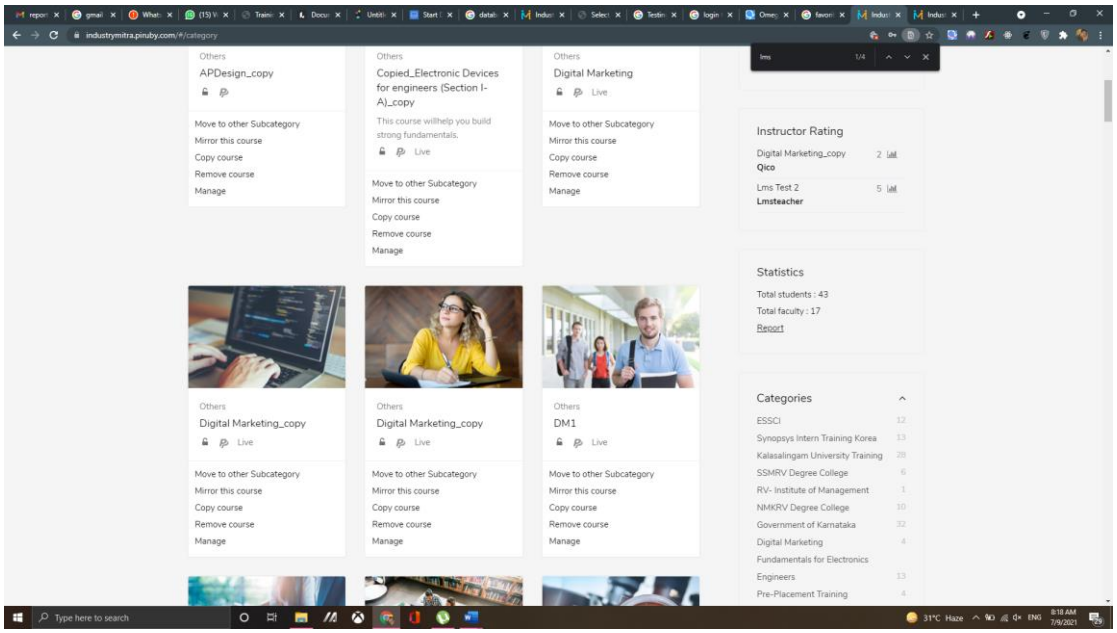

Fig 8.7.2 Check Ratings

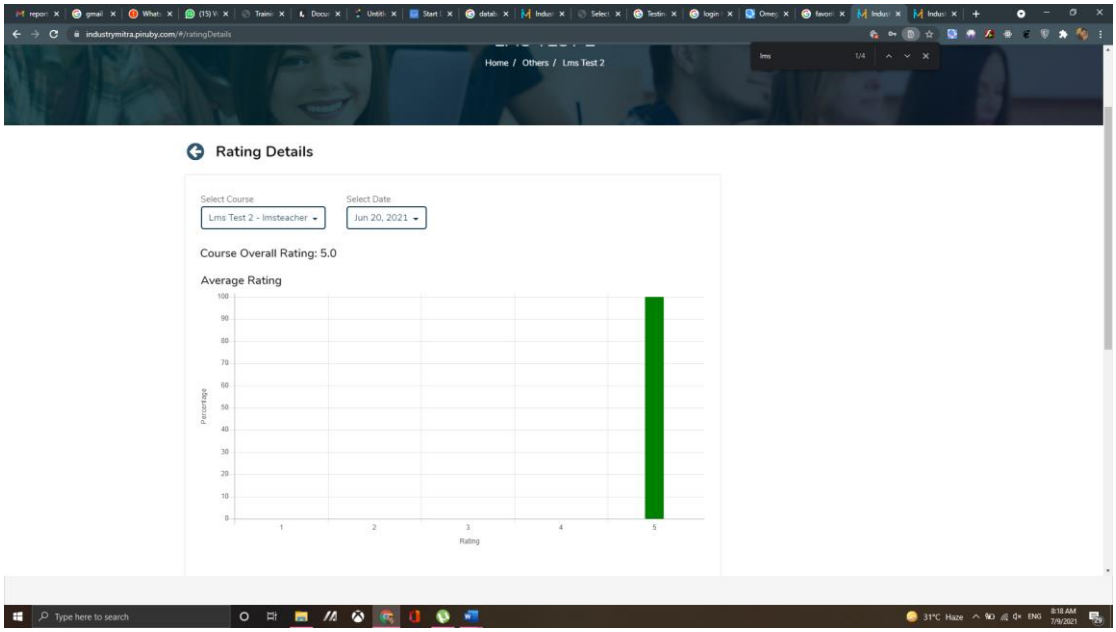

Fig 8.7.3 Check Rating in graph Form

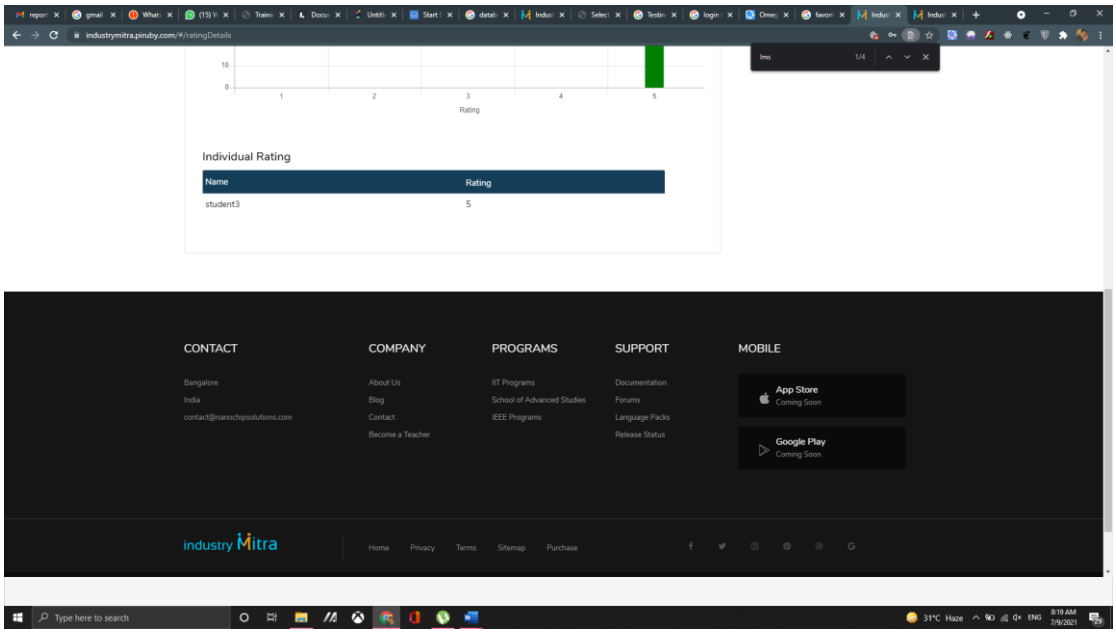

Fig 8.7.4 Check Individual Rating

REFERENCES

[1] JavaScript Docs

https://developer.mozilla.org/en-US/docs/Web/JavaScript

[2] Beginner's guide to HTML by S.K. Sahoo, 1999.

http://www.beginnerguide.com/html/allconcepts.html

[3] Complete CSS Guide

Maxine Sherrin and John Allsopp-O'Reilly Media; September 2012

[4] Angular JS Docs

https://docs.angularjs.org/guide

[5] Stack Overflow

www.stackoverflow.com

[6] Socket.io

https://socket.io/docs/v4

[7] W3 Schools

https://www.w3schools.com/

STUDENT PROFILE

Name: Gaurav Kataria

University Roll No: 17020004022

Class Roll No: CSE/17/022

Branch: Computer Science

Email Id: gkataria987@gmail.com

Phone No: 8178696212

SKILLS

- **Scripting Language**: JavaScript
- **Operating System**: Ubuntu, Windows
- **Programming Language**: C, C++, Java, Python
- **Database Management System**: - MS- SQL, Mongo DB
- **Other Tools and Technologies**: Angular js, React js, Html, CSS, Bootstrap, jQuery, Firebase

Github link: - https://github.com/Gaurav453

LinkedIn Profile: https://www.linkedin.com/in/gaurav-kataria987/

Code chef: - https://www.codechef.com/users/gaurav453/