INTERNSHIP REPORT ON

LEARNING MANAGEMENT SYSTEM

(Feb 2022 - June 2022)

Internship report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology in

Computer Science and Engineering

BY

NANDAN AGARWAL

ENROLL NO: 181235

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

TABLE OF CONTENT

DECLARATION

I hereby declare that this submission is my own work carried out at Testbook Pvt Ltd, Mumbai from February 2022 to June 2022 and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

SUBMITTED BY:

Nandan Agarwal
181235
Computer Science & Engineering and Information Technology Department.
Jaypee University of Information Technology, Waknaghat, Solan

CERTIFICATE

This is to certify that the work which is being presented in the project report titled **"Maintenance Learning Management System"** in the fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat, Solan is an authentic record of work carried out by **Nandan Agarwal (181235)** under the supervision of **Mr. Parv Tiwari (SDE Manager)**, Testbook Pvt Ltd, Mumbai, India.

NANDAN AGARWAL 181235

The above statement made is correct to the best of my knowledge.

Mr. Parv Tiwari
Lead FullStack Developer
Testbook Pvt Ltd.

Dr. Ruchi Verma
Associate Professor (Grade - I) (Dept. CSE)
Jaypee University of Information Technology

ACKNOWLEDGEMENT

I would like to thank the Manager of HR in Talent Acquisition, of Testbook, Mumbai for allowing me to do an internship within the organization.

I also would like to thank Mr. Parvi Tiwari and all the people that worked along with me at Testbook, Mumbai for their patience and openness. They created an enjoyable working environment. I also would like to thank Mr. Anto Raja for mentoring me throughout my internship and helping me learn new concepts and technologies used in Angular development.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Mr. Pankaj Kumar, Training & Placement Coordinator of our college for the facilities provided to accomplish this internship. I would also like to thank the Head of our Department Dr.Vivek Kumar Sehgal and the faculty for teaching us the skills required for this internship. And special thanks to my mentor in the college Dr. Ruchi Verma who supported me throughout the whole and guided me to achieve the best.

Finally, I must acknowledge the constant support and patients of my parents.

Nandan Agarwal (181235)

Computer Science & Engineering and Information Technology Department.

Jaypee University of Information Technology, Waknaghat, Solan

LIST OF FIGURES

CHAPTER 01

INTRODUCTION

1.1 Introduction

An internship is a professional teaching moment that provides students with real-world experience in their field of study or career objectives. An internship allows students to gain new skills while also exploring and advancing their professions. This paper describes my current internship with Testbook in Mumbai. This internship report describes the actions that assisted me in meeting a number of my stated goals. For my internship as a Frontend developer maintaining their internal product, I was assigned the profile of Software Engineer. During the initial months of the internship, we were requested to brush up our skills and revise coding in Javascript language, this was followed by working on an onboarding project to create a basic CRUD Application with the guidance of the mentor. Following the completion of this learning procedure, I was assigned to maintain their Learning Management System product of the company. This mainly involved resolving critical bugs followed by feature addition in the existing platform. I was also asked to add a new module in the LMS web titled **"Coding Module"** which would facilitate thousands of content creators to add a Coding Module into the existing courses which are there for the students who are preparing for the government exams. I was responsible for formulating the architecture and writing clean code which supports scalability.

1.2 Job Description

Software engineering is the application of engineering ideas to the conception, implementation, and technical administration of the software. Software engineering was created to solve the issues associated with low-quality software efforts. When software surpasses deadlines, budgets, and quality expectations, problems develop. It ensures that the software is created consistently, correctly, on schedule, at a reasonable cost, and in accordance with the specifications. Software engineering has become critical in order to keep up with the

quickly changing demands of users and the context in which the program is supposed to operate.

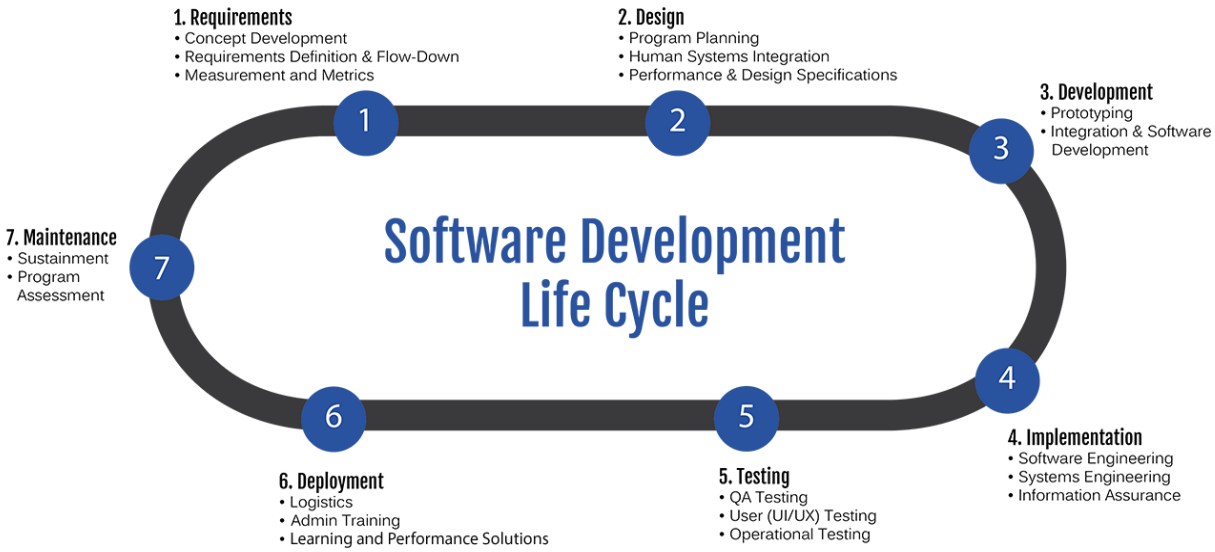

Fig 1.1: Software development cycle

The Software Engineer Trainee is in charge of assisting with software design and development. They collaborate with other team members to provide safe and trustworthy software solutions. This position's tasks and responsibilities are Application development (code, programming), Code debugging and testing New software applications, diagnosing, and resolving a wide range of technical obstacles collaborating with top executives identifying concerns, and providing solutions

1.3 Problem Statement

To develop and add a new module (Coding Module) in the existing web application (LMS) keeping in mind the best practices and ensuring scalability.

1.4 Objective

To add a feature into the existing web application supporting CRUD operations, with a scalable architecture which is easy to maintain as well as test, keeping in mind about the best practices. This internship gives freshmen experience in order for them to understand more about the industry. The goal was to learn enough so that we could work easily on the project that had been allocated to us.

1.5 AngularJS Development

AngularJS was an open-source front-end web framework for constructing single-page apps that was built on JavaScript (now deprecated). It was primarily managed by Google and a group of individuals and businesses.

By providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, as well as components typically used in web applications and progressive web apps, angularJS aimed to simplify both the building and testing of such applications.

The MEAN stack, which consisted of MongoDB database, Express.js web application server framework, AngularJS itself (or Angular), and Node.js server runtime environment, employed AngularJS as the frontend.

Google no longer updates AngularJS to fix security, browser compatibility, or jQuery vulnerabilities as of January 1, 2022. The Angular team recommends updating to Angular (v2+), but they have gave several other options.

The AngularJS framework read the HTML page first, which contained additional special HTML characteristics embedded in it. These characteristics were processed by Angular as directives to bind input and output elements of the page to a model represented by conventional JavaScript variables. Those JavaScript variables' values could be set manually in the code or retrieved from static or dynamic JSON re sources.

Declarative programming should be used to design user interfaces and connect software components, while imperative programming is better suited to establishing an application's business logic, according to AngularJS. The framework used two-way data-binding to automatically synchronize models and views, adapting and extending traditional HTML to show dynamic content. As a result, in order to improve testability and efficiency, AngularJS de-emphasized explicit Document Object Model (DOM) manipulation.

AngularJS's design goals included:

- To decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.
- To decouple the client side of an application from the server-side. This allows development work to progress in parallel and allows for reuse of both sides.
- To provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

AngularJS implemented the MVC pattern to separate presentation, data, and logic components. Using dependency injection, Angular brought traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server could be reduced.

AngularJS Example

```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>

<div ng-app="">
  <p>Name : <input type="text" ng-model="name"></p>
  <h1>Hello {{name}}</h1>
</div>

</body>
</html>
```

Fig 1.2: AngularJS Example Code (2 way data binding)

The core features of AngularJS are as follows −

- The automated synchronization of data between model and display components is known as data-binding.
- These are objects that relate to the model's scope. They serve as a connection between the controller and the view.
- Controllers are JavaScript functions that are tied to a certain scope.
- AngularJS has various built-in services, such as $http, which may be used to create XMLHttpRequests. These are singleton objects that are only used once in the programme.
- Filters return a new array after selecting a subset of elements from an array.
- Directives are DOM elements like as elements, attributes, css, and more that serve as markers. These may be used to make custom HTML tags that can be used as new widgets. AngularJS includes directives like ngBind, ngModel, and others.
- The displayed view containing information from the controller and model is called a template. These can be a single file (such as index.html) or partials that combine numerous views into one page.
- Routing is the notion of changing perspectives.
- Model View Whatever (MVW) is a design pattern that divides an application into three components: Model, View, and Controller, each with its own set of responsibilities. AngularJS implements MVC in a way that is more akin to MVVM than standard MVC (Model-View-ViewModel). Model View Whatever is how the Angular JS team refers to it.
- Deep linking allows the state of an application to be encoded in the URL so that it may be bookmarked. The programme may then be returned to its original state using the URL.

AngularJS MVC Architecture

Model View Controller is the abbreviation for Model View Controller. It's a web application development software design pattern. It's popular because it separates the application logic from the user interface layer, allowing for a clearer separation of responsibilities.

The three pieces of the MVC pattern are as follows::

1. **Model:** It is in charge of the application's data management. It updates itself in response to requests from the view and instructions from the controller.
2. **View:** View is in charge of showing all or a portion of the data to the users. It also defines the data in a certain format as a result of the controller's choice to provide the data in that manner. They are script-based template systems like JSP, ASP, and PHP that are simple to combine with AJAX.
3. **Controller:** It is responsible to control the relation between models and views. It responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

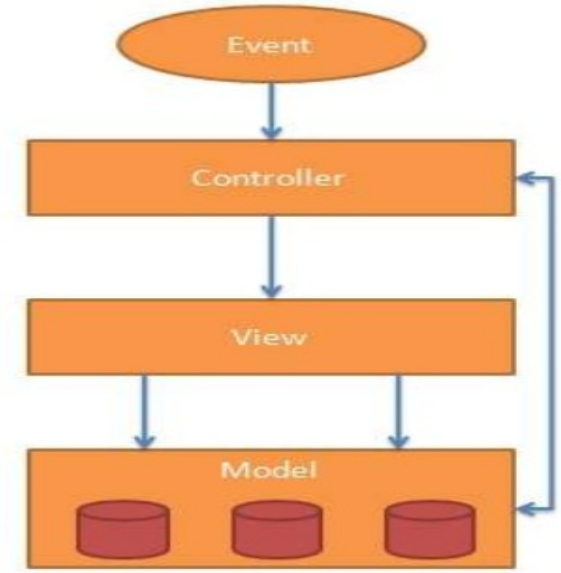

Fig 1.3: MVC Architecture (AngularJS)

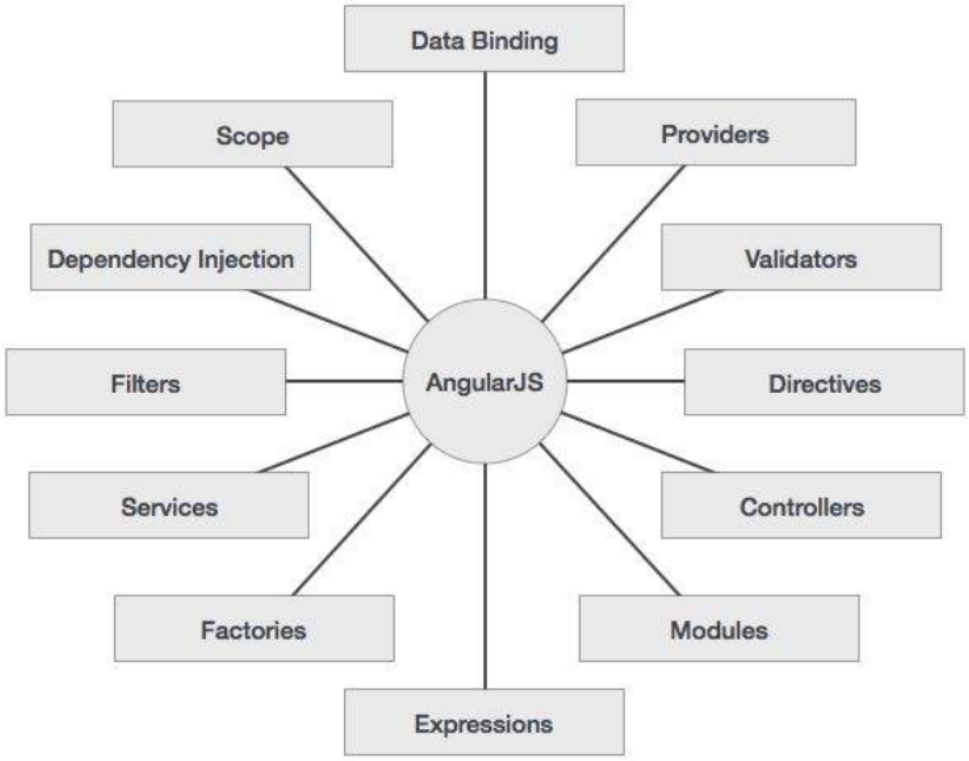

Fig 1.4: Overview of AngularJS Framework

1.6 Angular Development

Angular (also known as "Angular 2+" or "Angular CLI") is a Typescript-based free and open-source web application framework developed by Google's Angular Team and a community of individuals and businesses. Angular is a complete rebuild of the AngularJS framework by the same team that created AngularJS.

Angular is used as the frontend in the MEAN stack, which contains the MongoDB database, Express.js web application server framework, Angular (or AngularJS), and the Node.js server runtime environment.

Angular JS is an open source framework built on JavaScript. It was made by Google's programmers. This framework was intended to assist in the resolution of issues that might occur while dealing with single-page apps. When creating the framework, testing was also taken into consideration. The framework was created with testing in mind. In October of 2010, the framework was initially released.

Following are the key features of Angular 2 −

- **Components** − Angular's initial concentration was on controllers, although this has since switched to components rather than controllers. Components allow programmes to be divided into various parts. This makes it easier to maintain the application over time.
- **Typescript** − the newer version of Angular is based on TypeScript. This is a superset of JavaScript and is maintained by Microsoft.
- **Services** − A service is a collection of code that may be shared by several components of an application. So, if you had a data component that drew data from a database, you could turn it into a shared service that several apps could use.

In addition, Angular 2 has better event-handling capabilities, powerful templates, and better support for mobile devices.

Angular 2 has the following components −

- **Modules** − This is used to break up the application into logical pieces of code. Each piece of code or module is designed to perform a single task.
- **Component** − This can be used to bring the modules together.
- **Templates** − This is used to define the views of an Angular JS application.
- **Metadata** − This can be used to add more data to an Angular JS class.
- **Service** − This is used to create components which can be shared across the entire application.

CHAPTER 02

COMPANY DESCRIPTION

2.1 Testbook

Testbook is the most rapidly growing EdTech start-up, and it is positioned to change the sector. Testbook has raced to the front and is perfectly poised to conquer larger markets, with a registered user base of over 1.1 crore students, 370 crore questions done on the Web, and a killer Android App. Testbook is the ideal talent incubator. When someone joins the Testbook family, they learn and conquer. One gets trained by the greatest mentors and eventually becomes an expert in their profession. That said, in our start-up, the freedom to pick which projects one works on, how and when one works on them, and what one wants to add to them is valued.

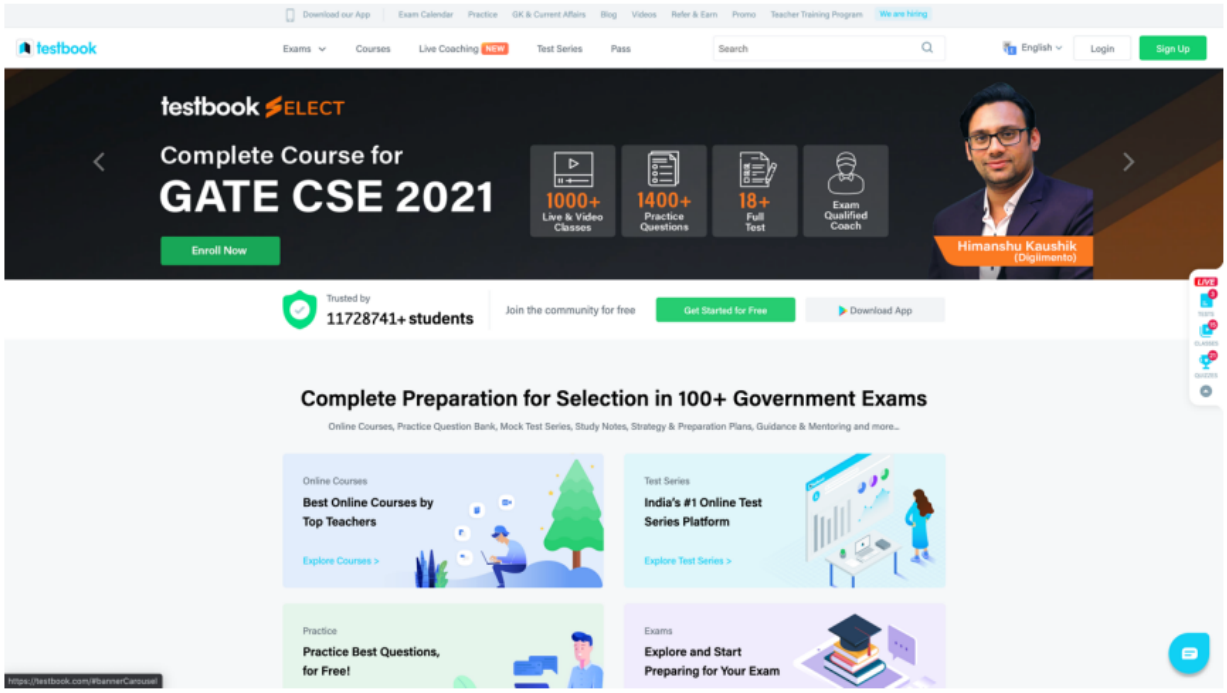

Fig 2.1: Testbook Main Web Application

2.2 Products

- **web-core:** The main web application, which is available to 15 lakh students studying for government examinations across India. AngularJS, Angular (2), React, and Wordpress are all used in the codebase.
- **LMS:** This is an internal product that the Testbook content team uses to plan and produce new courses for students. AngularJs and Angular are used in the codebase (13).

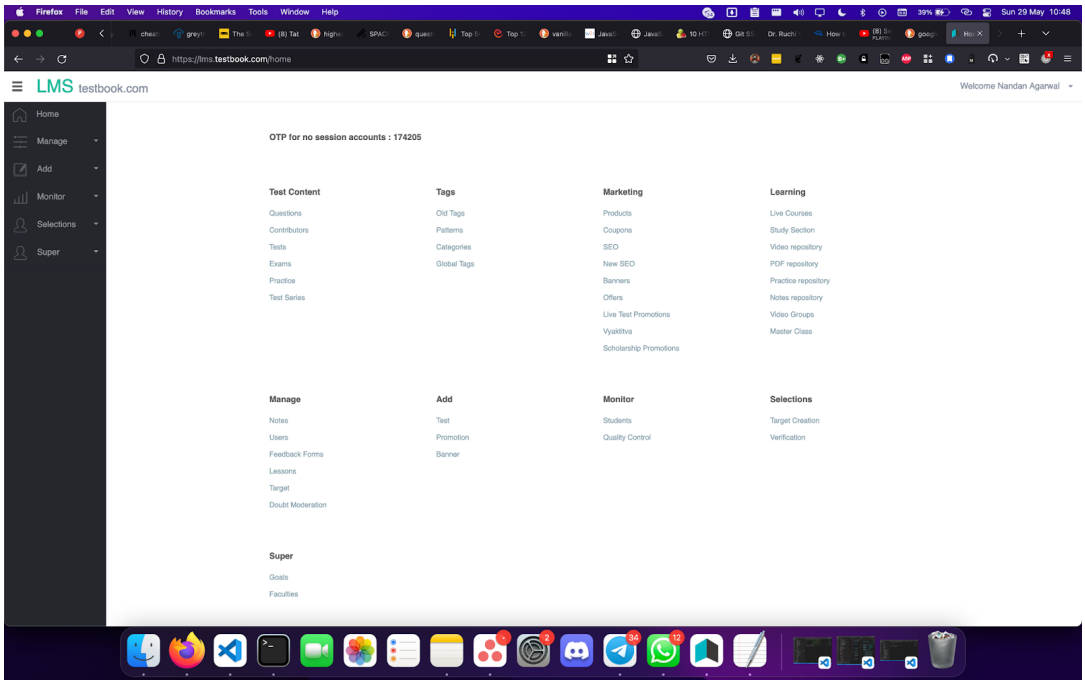

Fig 2.2: LMS Landing Page

- **tb-tool:** This is an internal product used by the Testbook Content team to stream live classes and serve thousands of students all over india. The codebase uses Angular (13) and WebRTC.
- **OGMA:** This is a new upcoming product which is currently in the beta phase. This product has been designed for all the job seekers. This is yet to be launched and released for the public. The codebase uses Angular (13).

CHAPTER 03

TOOLS AND TECHNOLOGIES USED

3.1 NodeJS

Node.js is a server-side stage in view of Google Chrome's JavaScript Engine (V8 Engine). Node.js was established by Ryan Dahl in 2009, and the latest variant is v0.10.36. In view of Chrome's JavaScript motor, Node.js is a structure for rapidly building quick and versatile organization applications. Due to its occasion driven, non-hindering I/O technique, Node.js is lightweight and productive, making it ideal for information serious continuous applications that stumble into various gadgets.

Node.js is a free and open-source cross-stage runtime climate for creating server-side and systems administration applications. Node.js programs are written in JavaScript and run on the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js likewise has a wide library of JavaScript modules, which makes creating Node.js web applications a lot more straightforward.

```
Node.js = Runtime Environment + JavaScript Library
```

Fig 3.1: Node Runtime Env

Features of Node.js :

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

- **Asynchronous and Event Driven** − The APIs in the Node.js library are all offbeat, or non-obstructing. It basically implies that a Node.js server won't ever hang tight for information from an API. The server continues on toward the following API in the wake of visiting one, and a warning framework called Events in Node.js helps the server in getting a reaction from the past API demand.

- **Single Threaded but Highly Scalable** − With a single threaded model, Node.js uses an event looping approach. Unlike traditional servers, which create limited threads to handle requests, the event mechanism allows the server to respond in a non-blocking and scalable manner. Node.js uses a single threaded application that can manage a far larger number of requests than traditional servers such as Apache HTTP Server.
- **Very Fast** − Being built on Google Chrome's V8 JavaScript Engine, Node js library is very fast in code execution time.
- **No Buffering** − Node.js applications never buffer any data. These applications simply output the data in chunks.

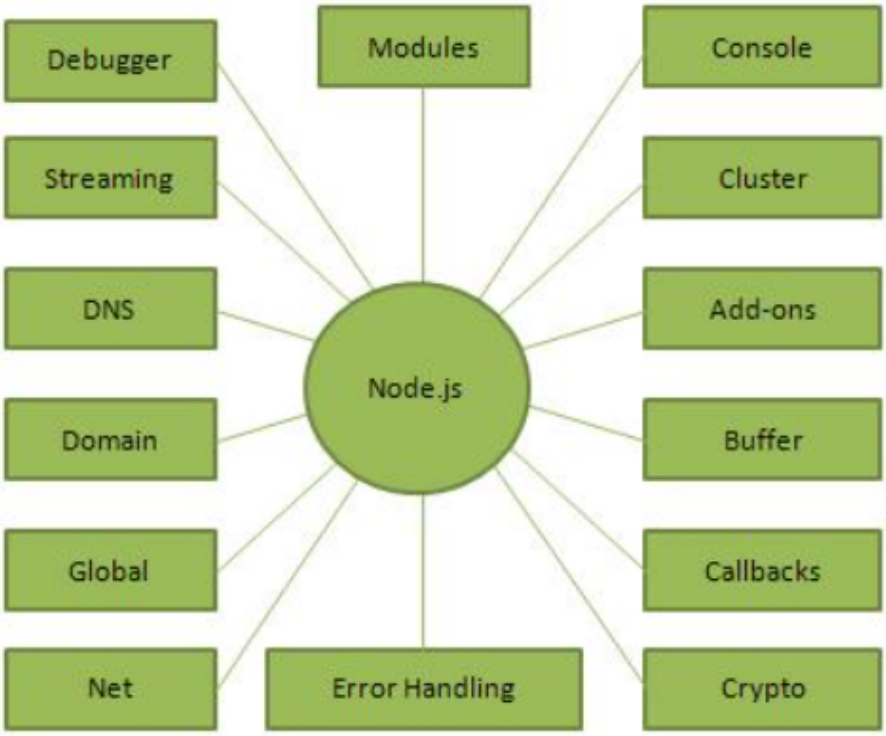

Fig 3.2: Overview of Node.JS Framework

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 Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications
- Data Streaming Applications

Where Not to Use Node.js?

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

3.2 Key Components Required

To kick start the development here are few of the main components which were used in order to facilitate easier development.

- Npm − this is known as the node package manager that is used to work with the open source repositories. Angular JS as a framework has dependencies on other components. And npm can be used to download these dependencies and attach them to your project.
- Git − This is the source code software that can be used to get the sample application from the github angular site.
- Editor − There are many editors that can be used for Angular JS development such as Visual Studio code and WebStorm. In our tutorial, we will use Visual Studio code which comes free of cost from Microsoft.

3.3 Github

Software developers and engineers may create public-facing cloud repositories for free using GitHub. You may add and alter files locally before "pushing" your changes back to the repository, where they will be available to the public, by downloading a GitHub repository to your device.

3.3.1 Repository

A repository is a part where all of the project's files are stored. Each project has its own repository, which may be accessed through a different URL.

3.3.2 Forking a Repository

You create a new project when you fork an existing one. This is an excellent tool for promoting the creation of new applications and initiatives. If you wish to contribute to a project on GitHub, fork it, make your changes, and then republish it as a new repository. You may easily apply modifications to your current fork if the repository from which you forked to build your new project changes.

3.3.3 Commiting the changes

When you fork an existing project, you are essentially starting from scratch. This is an excellent tool for incentivizing the creation of new applications and initiatives. If you wish to contribute to a project on GitHub, fork it, make your changes, and then republish it as a new repo. If the repository from which you forked to make your new project changes, you may update your existing fork fast.

3.3.4 Pull Requests

You've forked a repository, made a fantastic change to the project, and you'd like the original developers to notice it—and perhaps include it into the original project/repository. To accomplish this, create a pull request. The authors of the original repository can go over your work and determine whether or not to incorporate it in the project. When you submit a pull request, GitHub establishes a direct line of communication between you and the project's main maintainer.

CHAPTER 04

WORK DESCRIPTION

4.1 Onboarding Project

The goal was to create a simple Single Page Application (SPA) that performs CRUD operations utilizing the cutting-edge Angular Framework. The Mentor provided us a quick overview of how to use the various resources in order to complete the assignment.

A tech spec sheet was also prepared, which included a sprint timetable and a description of all the needed features and functionalities of the programme.

The complete application's flow was also considered and incorporated into the project.

4.1.1 Basic Features

- Search Functionality based on two binded search component
- Fetch of appropriate results from the IMDB api
- Paginated View of the matching results
- Descriptive Page for Movie

4.1.2 Tech Stack

This Project was basically developed in Angular 13 Framework. All the packages were installed using the Node Package Manager. (npm). Node version 16 has been used in this project.

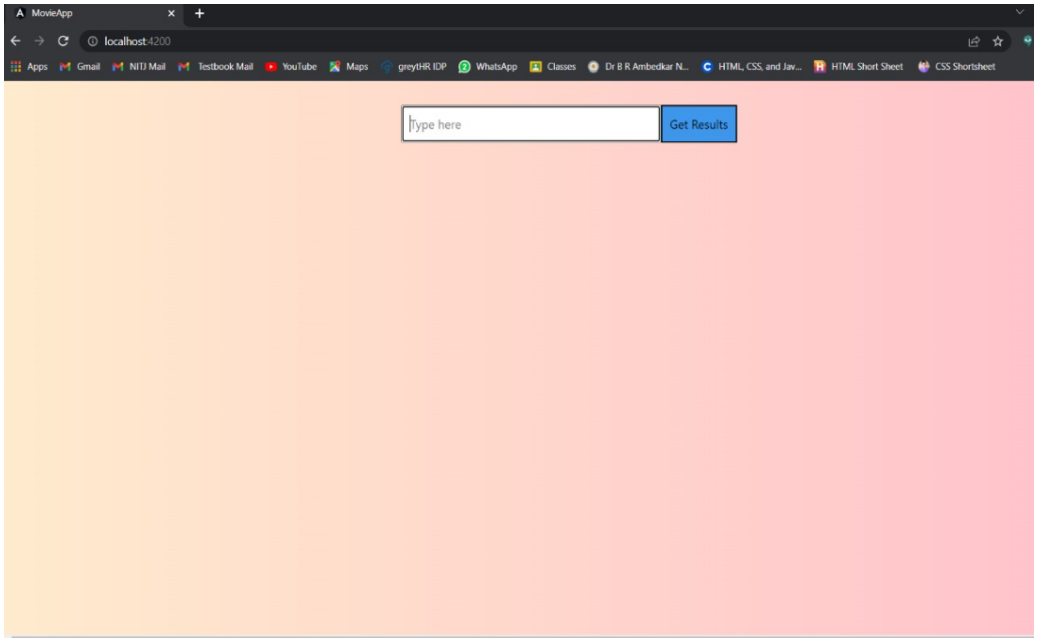

Fig 4.1: Search Functionality (Onboarding Project)

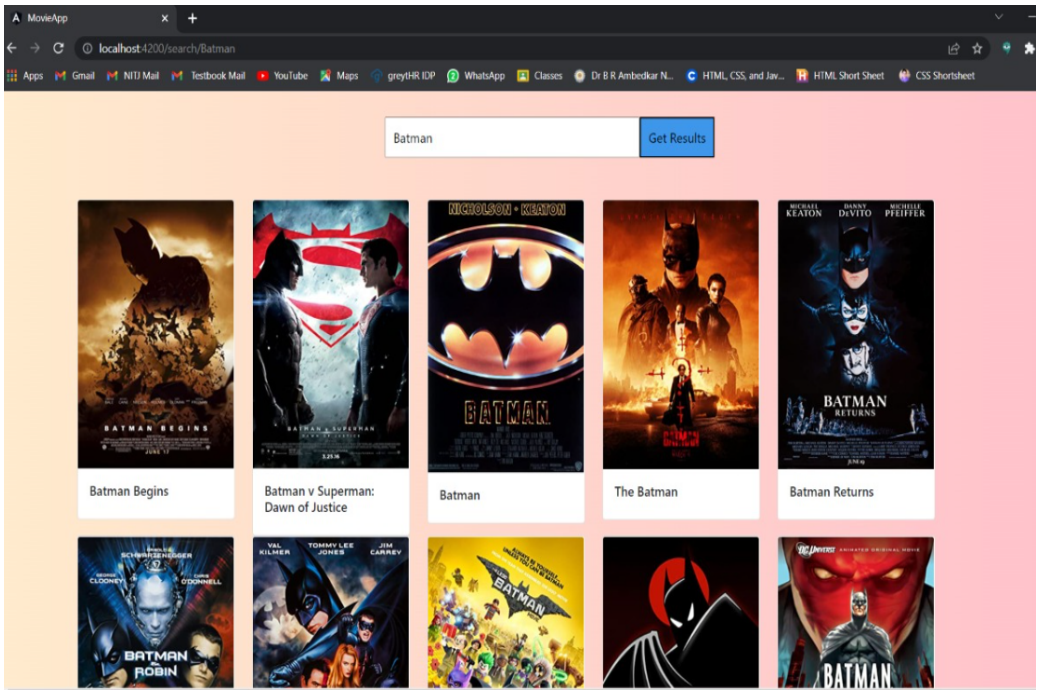

Fig 4.2: Display of movie results (Onboarding Project)

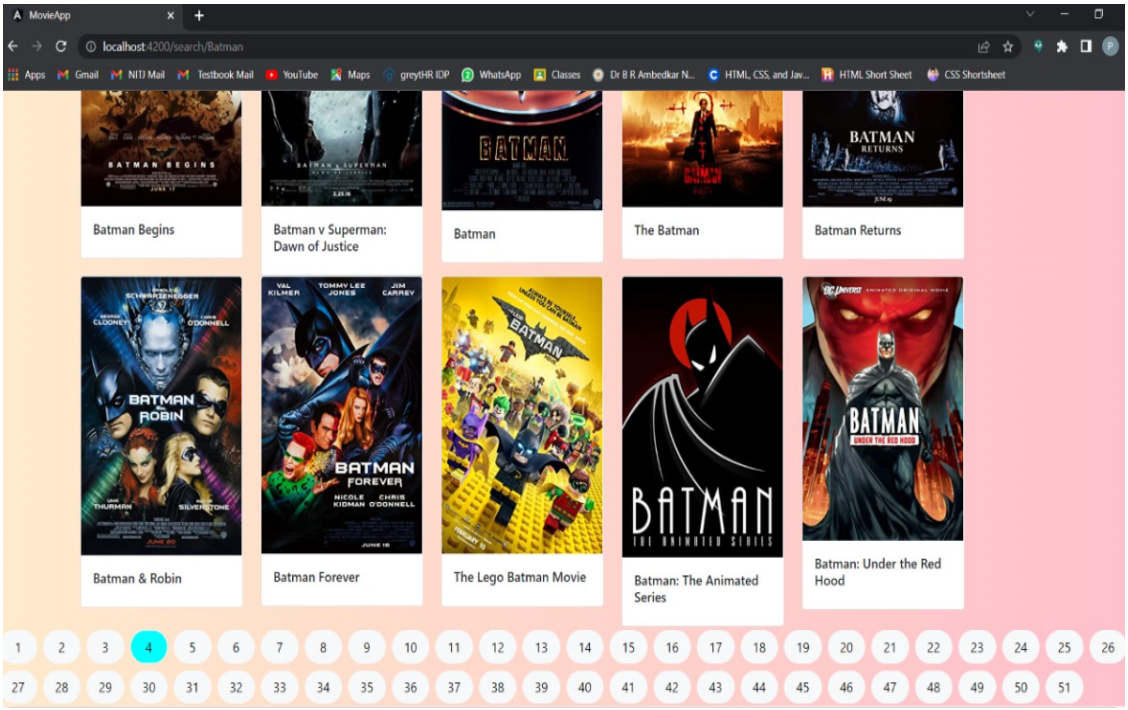

Fig 4.3: Paginated Component (Onboarding Project)

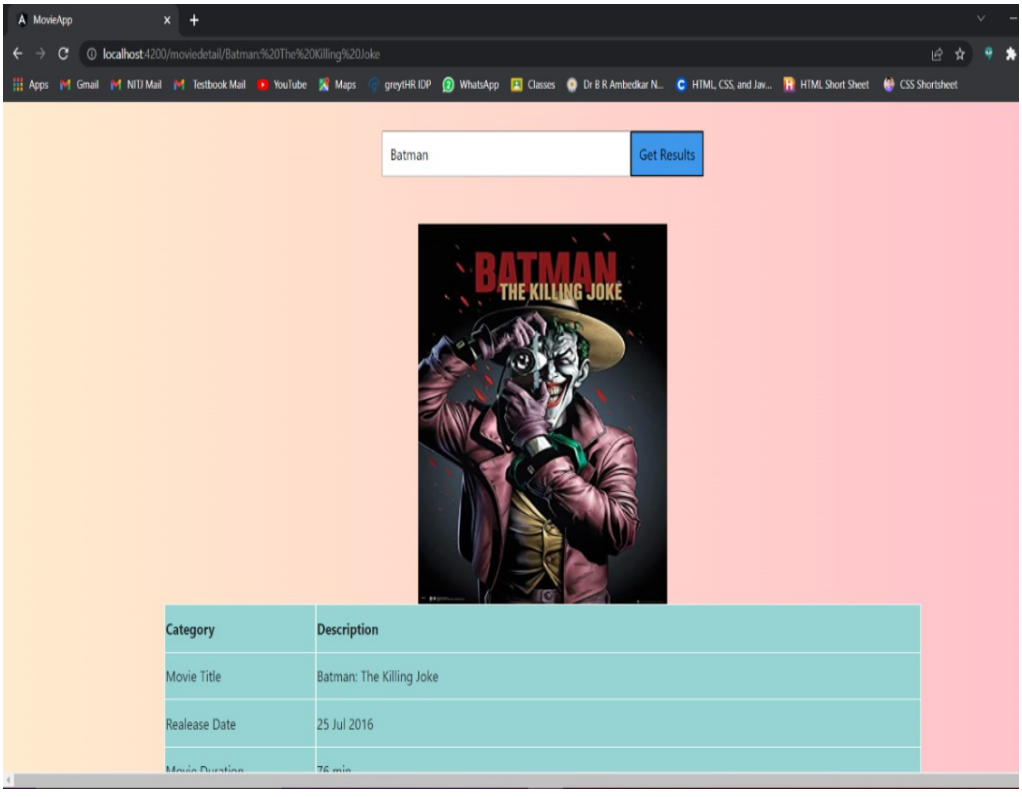

Fig 4.4: Movie personalized Description (Onboarding Project)

4.2 Addition of Features on LMS

Several features were introduced to the Learning management system throughout the internship to give greater functionality to the testbook content team while building courses. Here's a complete list of all the things I worked on and contributed throughout my time at the company.

4.2.1 Adding Coding Module

In the current learning pod, a new module, "Coding Problems," was introduced. This Coding Problem Repository gave the content author the option of generating and uploading code snippets to the course.
This Module comprised of 2 broad phases:

- Creation of New Pages to manage Coding Problem CRUD (Angular 13)
- Adding the feature to include Coding Problems in the existing sections of LMS Application (AngularJS)

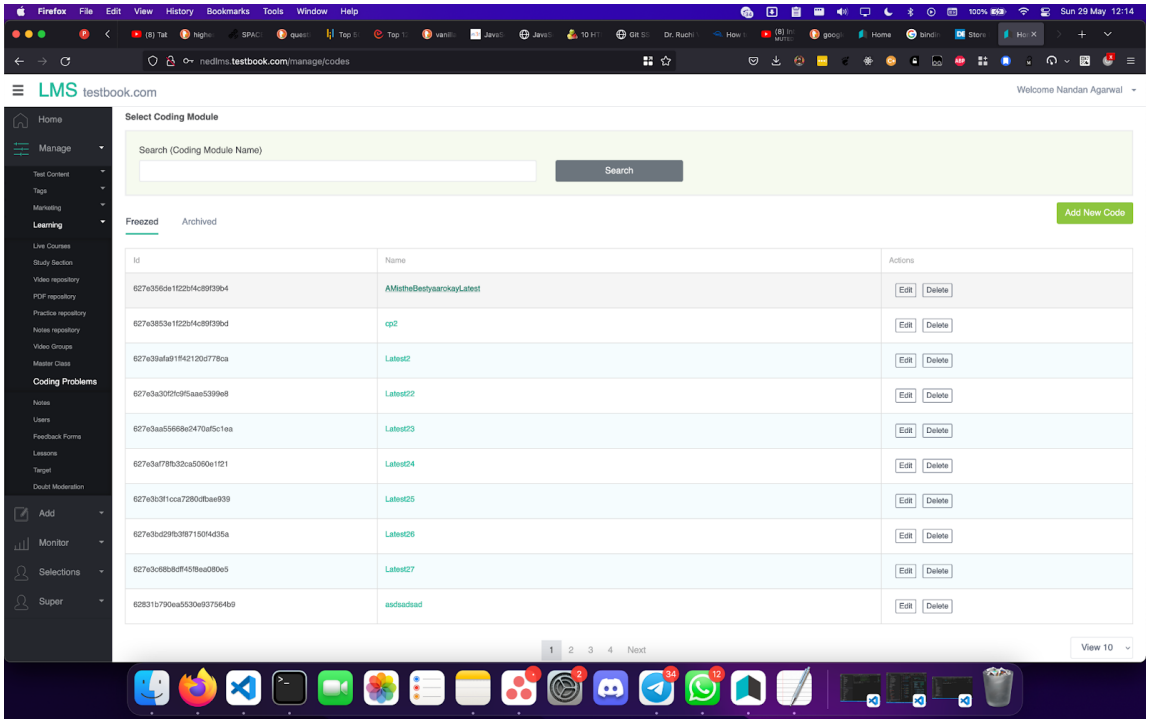

Fig 4.5: Managing Existing Coding Problems Paginated View

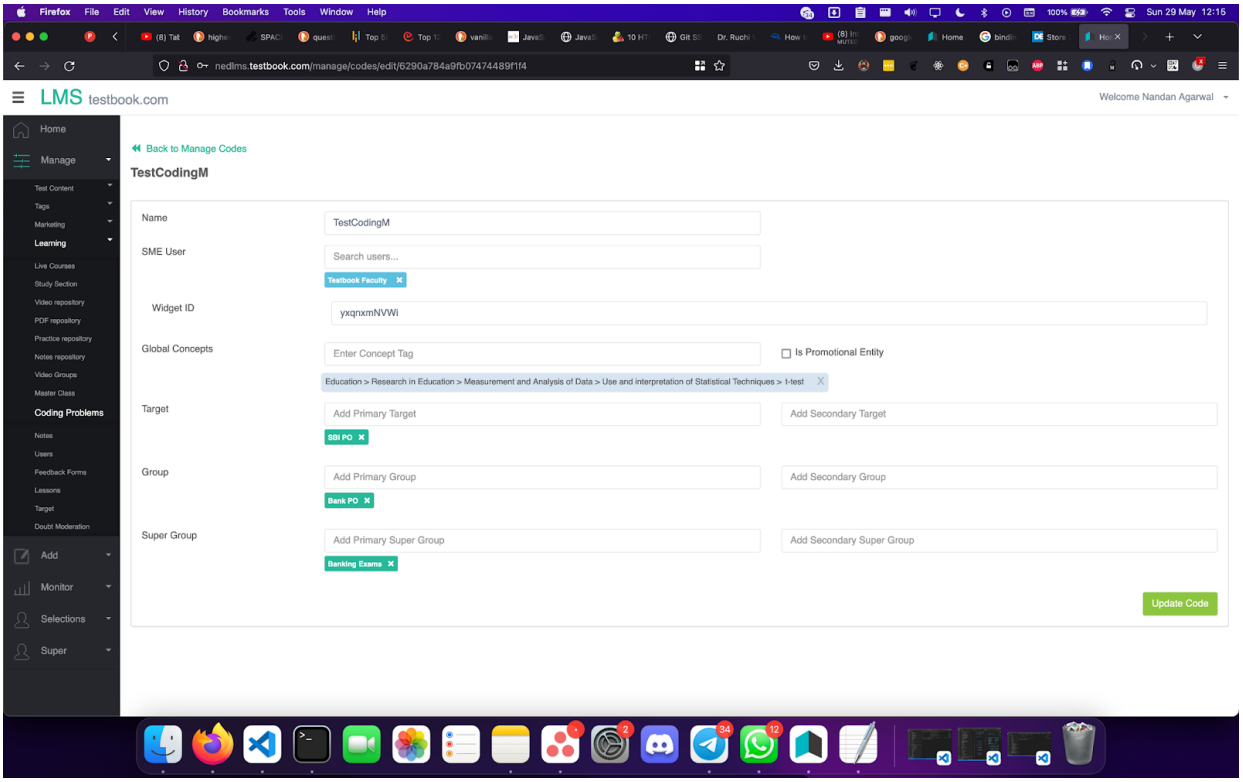

Fig 4.6: Add/Edit Coding Problem (Angular 13)

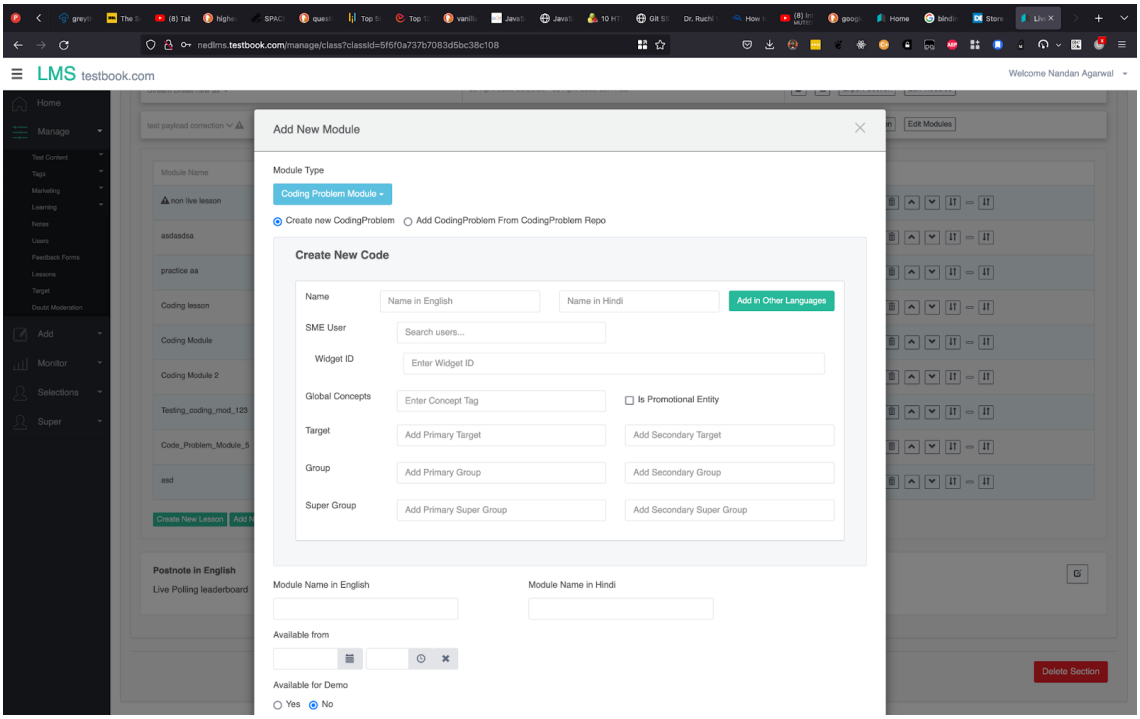

Fig 4.7 Addition of Coding Problem in the existing Live Course Section

4.2.2 MultiLang Support For Practices

This was one of the major inclusion in the LMS Product. This support was long overdue. The practice notes which the content team created lacked the multilanguage support. With this new addition Content Creators can now create practice courses in a variety of languages. This was written and developed in AngularJS

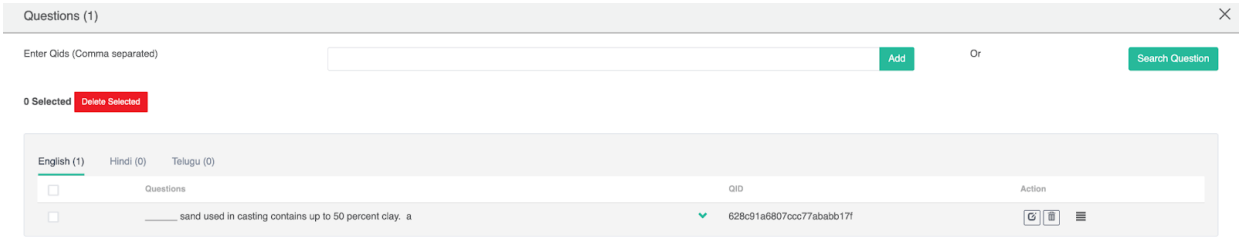

Fig 4.8 MultiLang Support Practice

4.2.3 Reported Question Logs

When any question is reported in any test, we need a log of it to help our content members to fix it.

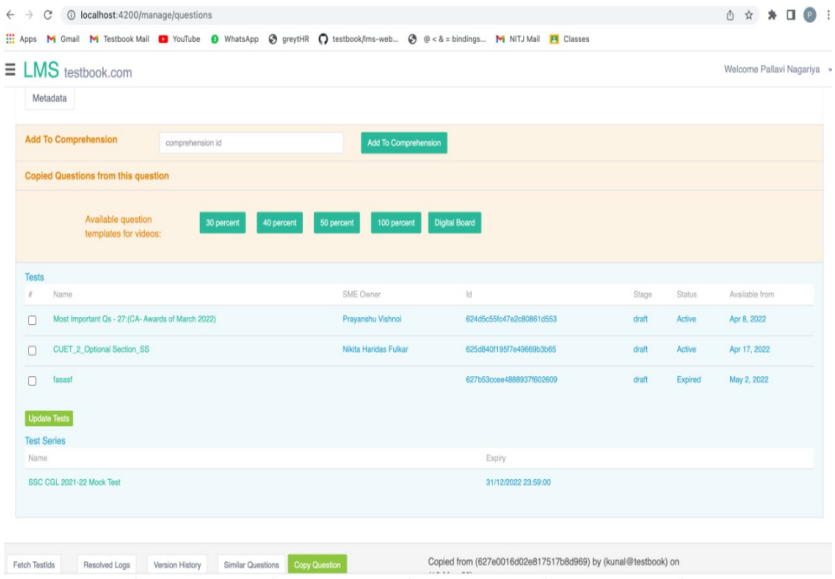

Fig 4.9 Reported Question Logs

4.2.3 Exam Date Inside Test Series

If any exam consists of stages then a stage-wise date setup will be shown to the user to work with it.

If not then a single stage setup is shown and on change of the target, if date setup needs to be changed, it'll be triggered.

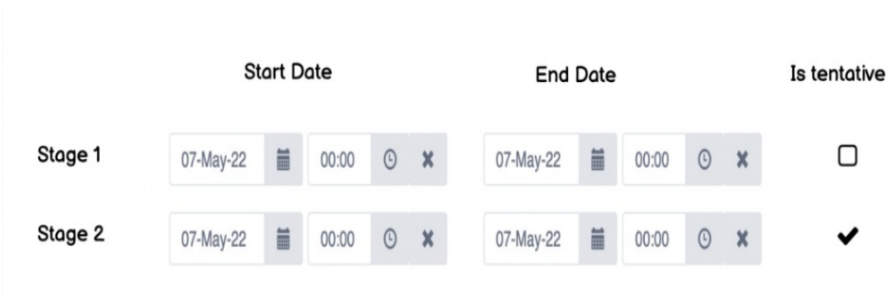

Fig 4.10 Exam Dates in test series

CHAPTER 05

CONCLUSION

5.1 Conclusion

Finally, I'd want to say that this internship has been a terrific and rewarding experience for me. I can honestly say that my time spent with the Testbook was quite beneficial. Needless to say, the technical components of my work aren't perfect, but with more time, they might be. As someone who had minimal prior experience with frontend programming in general, I feel the time I spent studying and understanding was well spent because it helped me create a fully usable app service. Two of the most significant things I've learned are time management and self-motivation.

5.1 Future Work

As a consequence, any future features and technologies I produce will have a significant influence on small enterprises by technologically elevating them and assisting them in quickly developing smart shops using touch. As a result, further work will be done in the future to provide consumers with a better self-checkout experience and a more accurate and up-to-date point-of-sale application.

5.2 Mentor’s Review

He performed admirably during training; he is a quick learner who takes less time to comprehend concepts. At the same time, he has exceptional problem-solving skills, making him an invaluable asset to our organization.

Parv Tiwari
Senior Software Engineer,

References

1. About Testbook https://testbook.com/
2. LMS Product https://lms.testbook.com/