

ZOPSMART MENTOR PRO

PREPARED BY

Mehul Khandelwal (2018BTechCSE127)

FACULTY GUIDE

Mr. Gaurav Raj



Department of Computer Science & Engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University, Jaipur

May 2022

ZOPSMART MENTOR PRO

Practice School-2

Submitted in partial fulfilment of the requirements
for the degree of
Bachelor of Technology in Computer Science Engineering

By:

Mehul Khandelwal (2018BTechCSE127)

Faculty Guide:

Mr. Gaurav Raj



Department of Computer Science & Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University, Jaipur

May 2022

CERTIFICATE

This is to certify that the Practise School-2 project work entitled “ZopSmart Mentor Pro” submitted by **Mehul Khandelwal (2018BtechCSE127)** towards the partial fulfilment of the requirements for the **Bachelor of Technology in Computer Science Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practise School-II examination.



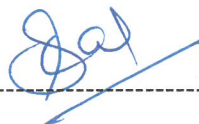
Mr. Gaurav Raj

Assistant Professor,

Department of Computer Science
Engineering

Institute of Engineering & Technology
(IET)

JK Lakshmipat University Jaipur



Dr. Sonal Jain

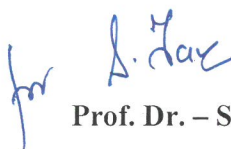
Co-Ordinator and Professor,

Department of Compute Science
Engineering

Institute of Engineering & Technology
(IET)

JK Lakshmipat University Jaipur

Date of Submission: 13th May 2022



Prof. Dr. – Sanjay Goel

Director

Institute of Engineering & Technology

JK Lakshmipat University

ACKNOWLEDGEMENTS

The internship opportunity I had with ZopSmart Technology Pvt. Ltd. was a great chance for learning and professional development. Therefore, I consider myself fortunate individual as I was provided with an opportunity to be part of ZopSmart. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period.

I perceive as this opportunity as a big milestone in my career development. I have taken efforts in completing this internship. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to express heartfelt gratitude to **Prof Dr.- Sanjay Goel**, Director, Institute of Engineering & Technology for providing this opportunity to me. Also, I would like to thank my faculty guide **Mr. Gaurav Raj** for his constant support and guidance.

I am highly indebted to **ZopSmart Technology Pvt Ltd.** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express our gratitude towards our parents & the members of ZopSmart Technology Pvt Ltd. for their kind co-operation and encouragement which helped me in the completion of this project and internship. I would like to express my gratitude and thanks to **Mr. Vikash Kumar, Mr. Ujjawal Misra, Mr. Shreyas R S, and Mr. Prashanth Kolakani** industry persons for giving me such attention and time.



Sincerely yours,
Mehul Khandelwal
(2018BtechCSE127)

ABSTRACT

Industrial training is necessary to build the gap between the theoretical knowledge and the practical knowledge to utilization of the concepts which is required in the corporate world. It is an important phase of a student's life. It provides a platform for the students to practice the concepts acquired in the classroom also it involved the work on live projects using advanced technologies which are used in industry. A well planned, properly executed and evaluated industrial training helps a lot in developing a professional attitude. It develops an awareness of industrial approach to problem solving, based on a broad understanding of process and mode of operation of organization.

The aim and motivation of this industrial training is to receive discipline, skills, teamwork and technical knowledge through a proper training environment, which will help me, as a student in the field of Information Technology, to develop a responsiveness of the self-disciplinary nature of problems in information and communication technology.

The ZopSmart MentorPro is an end to end mentoring web application with additional feature-set. The purpose of developing this application is to improvise the current practise of communicating between the two users i.e., the mentor and the mentee where irrespective of the subject set users will be getting timely guidance and necessary tutorials and sessions with the help of Database management system concepts and computer software, fulfilling their requirement, so that the valuable data can be stored for easy accessing and manipulation. It is a web-based application developed using Java powered by Java based framework spring boot for handling APIs at backend and MySQL as database whereas the frontend is developed using React.

LIST OF FIGURES

Figure	Page No
Figure 1.1 ZopSmart Technology	1
Figure 2.1 Use Case Diagram	5
Figure 2.2 Spring Boot Framework	13
Figure 2.3 User Credential Journey	14
Figure 2.4 Flyway	16

LIST OF TABLES

Table	Page No
Table 2.1 Login Use Case	6
Table 2.2 Admin Use Case	7
Table 2.3 Ban Revoke Use Case	8
Table 2.4 Report User use Case	9
Table 2.5 Invite user Use Case	10
Table 2.6 Schedule a video meet Use case	11
Table 4.1 Login Test	20
Table 4.2 Block Domain Test	21
Table 4.3 Invite user Test	21
Table 4.4 Signup Test	22
Table 4.5 Report User Test	22
Table 4.6 Ban/Revoke Test	23

CONTENTS

<i>CERTIFICATE</i>	<i>iii</i>
<i>ACKNOWLEDGEMENT</i>	<i>iv</i>
<i>ABSTRACT</i>	<i>v</i>
<i>LIST OF FIGURES</i>	<i>vi</i>
<i>LIST OF TABLES</i>	<i>vii</i>

CHAPTER 1: INTRODUCTION 1-3

1.1 About Zopsmart Technologies	1
1.2 General Introduction	2
1.3 ZopSmart MentorPro	2-3
1.4 Objectives	3

CHAPTER 2: PROPOSED SYSTEM 4-26

2.1 Problem Statement	4
2.2 Modules of Proposed System	4-5
2.2.1 Admin	4
2.2.2 User	4-5
2.3 Use Case Approach	5-11
2.3.1 Use Cases	6-11
1. Login Use Case	6
2. Basic Admin Use Case	7
3. Ban Revoke User	8
4. Report the user	9
5. Invite an user	10
6. Schedule a video meet	11
2.4 Technologies Used	12-17
2.4.1 Next JS	12
2.4.2 Spring Boot	12-13
2.4.3 Spring Security	13-14
2.4.4 Docker	14-15
2.4.5 Github	15
2.4.6 Github Actions	15-16
2.4.7 Database Migrations	16-17
2.4.8 Unit Testing	17

CHAPTER 3: SOFTWARE DESIGN	18-19
3.1 Design Strategy	18
3.1.1 Function oriented design	18
3.2 Feasibility Study	18-19
2.2.1 Technical Feasibility	18-19
3.2.2 Time Schedule Feasibility	19
3.2.3 Operational Feasibility	19
3.2.4 Implementation Feasibility	19
 CHAPTER 4: CONSTRUCTION	 20-23
4.1 General Introduction	20
4.2 Test Scenarios	20-23
4.2.1 Login	20
4.2.2 Block Domain	21
4.2.3 Invite User	21
4.2.4 Signup	22
4.2.5 Report User	22
4.2.6 Ban/Revoke User	23
 CHAPTER 5: CONCLUSION	 24
REFERENCES	25

CHAPTER 1: INTRODUCTION

1.1 ABOUT ZOPSMART TECHNOLOGY

ZopSmart Technology is a cutting edge retail technology company that provides you all the tools to build your e-commerce business. Whether you are a conventional retailer intending to build your omni-channel business or an online-only retailer intending to build your e-commerce business, we have a suite of products that will quickly and easily empower you to achieve your goals. They envision to be the #1 retail technology platform of the world. Their expertise in the retail domain is built on our team's immense experience in the field of FMCG, retail, supply-chain and logistics. All our experience has been baked into business processes that are embedded in our products. Their solution has been built over 7 years and is handling thousands of transactions every day. Our suite of products are one of the most advanced, stable and scalable solutions available in the world of retail technology. Their solutions include e-commerce, m-Commerce, Digital Marketing, Customer Management, e-wallet and Order Management etc.



Figure 1.1 ZopSmart Technology

Using e-Commerce they give customers a world-class shopping experience - Intuitive search to quickly find products. Personalized product listing for ease of ordering. Self-service rescheduling and returns functionality for frictionless order experience. M-Commerce provide your customers a superlative shopping experience on mobile devices - Responsive website that adapts to all screen sizes. Native Android and iOS mobile apps for world-class experience. Persistent cart across devices for convenience. E-Wallet use integrated wallet to build customer loyalty - Integrated wallet to enable quick and hassle-free checkout. Refunds and cashback to wallet to encourage repeat purchase. Wallet-credit as gift-cards for acquiring of new customers at low cost. Customer Management manage your online as well as offline customers easily and efficiently - Single interface to view all your customers with their contact details. Available details of each customer's purchase history, orders and loyalty points. Option to contact the customer from the interface by phone-call, sms and e-mail.

1.2 GENERAL INTRODUCTION

Around the world, growth of own self is the most important aspect to survive in the fast growing world and to be more productive in every aspect of your work, whether personal or professional. Mentorship is something which is often required by people to learn and grow in various fields and phases of life be it personal mentoring sessions or professional. Mentorship Programs and helps in accelerating the personal and professional development of the mentees. This is achieved by providing mentees with guidance, advice and feedback from mentors with more experience than themselves. To ensure the growth of every individual, it is necessary to have a straight and both way end to end communication where if at a time oneself is acting a mentor, at some other point of time, he can be acting as a mentor to learn.

The benefits of mentoring for both mentors and mentees are related to what an organization would want but are understandably more tied to their own growth rather than the company. For example, a mentee may want to become more engaged in their work by talking about their goals and aspirations with a mentor. The company may desire this as well because a more engaged employee will do better work.

Being a mentor is a great opportunity for senior leaders or experienced employees to give back to the next generation of the company. It's also a valuable learning experience that reveals their communication skills and ability to build relationships with others. Ultimately, mentors become more competent as leaders and communicators as they guide and help rising talent. The goals of a mentee are central to the success of the mentoring relationship. What do they want to get out of meeting with a more experienced leader once or twice a month? Some may want to gain visibility from leadership so they'll be considered for new opportunities. Other mentees may want to learn specific hard or soft skills like communication, negotiation, or conflict management.

1.3 ZOPSMART MENTORPRO

ZopSmart MentorPro is an end to end mentoring web application with additional feature-set. The purpose of developing this application is to improvise the current practise of communicating between the two users i.e., the mentor and the mentee where irrespective of the subject set users will be getting timely guidance and necessary tutorials and sessions with the help of Database management system concepts and computer software, fulfilling their requirement, so that the valuable data can be stored for easy accessing and manipulation.

The purpose of this system is to provide professional and nonprofessional guidance to the users irrespective of their area, and help the mentees to assign mentors for rectifying their doubts on a respective topic or subject. This system consists of different sets of features which vary as per the role of the user. The system contains admin, mentee and mentor roles.

Zopsmart MentorPro allows the users to look for the experts or mentors based on different search criteria such as area of expertise, domain, experience, availability etc. where they can schedule a session with them and have a one to one session either in person or online and keep a track of the pre-session, in-session and post-session notes. Users can also invite their fellow mate on the platform, have an ability of raising issues and giving feedback and managing

profiles, etc. Moreover, mentors can keep a track of their mentees and works assigned and can have also have a normal chat session.

Hence, it enables the smooth functioning and growth of a healthy relationship between the mentor and the mentee. It is a web-based application developed using Java powered by Java based framework spring boot for handling APIs at backend and MySQL as database whereas the frontend is developed using React.

To achieve the effective performance different Database strategies are implemented. User can access the portal using either normal email IDs or through social logins such as google, Facebook and LinkedIn, after which he will be taken to the on boarding page where they will be able to fill their personal, professional and educational details.

1.4 OBJECTIVE

This system aims at solving the problem of some traditional systems where it was difficult to handle the mentee records and give timely feedback to the mentees. For the users of an organization, who need assistance regarding any technical and nontechnical queries, this system provides online communication with mentors in an easy and effective manner which saves time and energy. It will help to develop a positive relationship between both the parties.

CHAPTER 2: PROPOSED SYSTEM

2.1 PROBLEM STATEMENT

Around the world, growth of own self is the most important aspect to survive in the fast growing world and to be more productive in every aspect of your work, whether personal or professional. Mentorship is something which is often required by people to learn and grow in various fields and phases of life be it personal mentoring sessions or professional.

This is achieved by providing mentees with guidance, advice and feedback from mentors with more experience than themselves. To ensure the growth of every individual, it is necessary to have a straight and both way end to end communication where if at a time oneself is acting a mentor, at some other point of time, he can be acting as a mentor to learn.

Hence, there is a need of system where users can come and irrespective of their domain and expertise can look for mentorship and guidance and have a track of their progress throughout the journey.

2.2 MODULES OF PROPOSED SYSTEM

Currently the ZopSmart MentorPro has two modules namely Admin and User, both varying in the terms of authority to manipulate the information available on the portal.

2.2.1 Admin

Admin is responsible for configuration and dependable functioning of the automobile portal. Following are the sub modules of Admin:

1. **Logs of all the user sessions:** How many users have logged into the application based on certain search criteria (one month, week, day, etc)
2. **User sign-up stats:**
 - Show the list of all the users who have signed up by the date (filtering / sorting ability should be there)
 - How many invites have been sent by existing users and how many of them signed up with those invites.
3. **Logs of all the meetings ever created:** The admin should be able to see the list of meetings scheduled based on certain search criteria (one month, week, day, etc)
4. **Controls for FAQ page:**
 - Adding/removing a question along with the answer.
 - Hide/unhide questions with a toggle on every added question.
5. **Controls for Quick tour slides:** Increasing/ decreasing the number of slides. Adding Images/ text to those slides. Enable/disable quick tour.
6. **Controls to ban /revoke ban any user.**
7. **Feature to send an email notification/ platform notification to all users.**
8. **Controls to edit the email contents.**
9. **Feature to block domains for new users signing up.**
10. **Feature to see and ban (take action on, if needed) the reported users.**

2.2.2 User

Users will be the one who will be interacting with the platform directly and will have to access to various functionalities on the system

- a) **Login/Signup:** users will be signing up and the log in into the system where appropriate dashboard will open as per the role.
- b) **Onboarding Page:** users will be giving their inputs about the personal, professional and educational details which they can edit later on.
- c) **Mentor Session:** users can schedule a meet with their respective mentors and sent the invite link for the meet sessions.
- d) **Manage Profile:** user can look after their own profiles and play with the information along with visibility status of the information.
- e) **Referrals:** users will be able to send the invites using emails of their mates or the invite link through various social platforms.
- f) **Report/Block User:** users can report and block users as per their choice in case of any nuisances.

2.3 USE CASE APPROACH

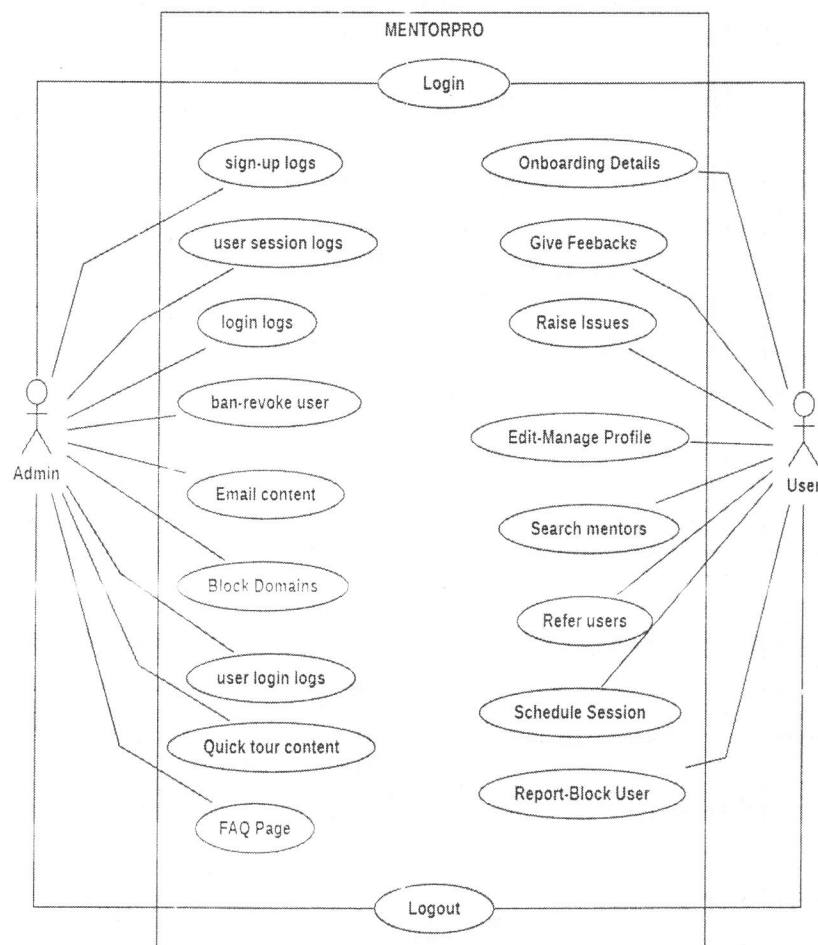


Figure 2.1 Use Case Diagram

2.3.1 USE CASES:

1) Login Use Case:

Table 2.1 Login Use Case

1	Use Case	Log in
2	Primary Actors	User
3	Secondary Actors	UI, System, Database
4	Pre-Condition	1. The user is already on the login page. 2. The user should be registered on the platform.
5	Post-Condition	1. The user can successfully login into the platform.
6	Main Success Scenario(Main Flow)	1. The user enters his Login ID in the Login ID text field. 2. The user enters the password in the Password field. 3. The user clicks on the Login button. 4. The System will validate the details. If validated the user will be successfully logged in.
7	Extensions1(Alt. Flow) Wrong Login ID, Password	5. If the system failed to validate the credentials, The user will stay on the same page. 6. The user will be shown a message - <i>"Please, enter the correct login ID and password."</i>
8	Extensions2(Alt. Flow) Wrong Password	1. The user enters the correct username. 2. The user tries to log in with the wrong password twice. (he has received the same warnings twice <i>"Please, enter the correct username and password."</i> 3. After the user tries to log in with the wrong password for the third time, He will be shown a message by the system - <i>"Please enter the correct username and password. If you want to reset the password, please click on the Forgot password button."</i>

2) Basic Admin Use Case:

Table 2.2 Admin Use Case

1	Use Case	Admin page
2	Primary Actors	Admin
3	Secondary Actors	UI, System, Database
4	Pre-Condition	1. The Admin already has the login URL. 2. The admin is on the admin login page.
5	Post-Condition	1. The admin is able to use various features on the Admin page.
6	Main Success Scenario(Main Flow)	1. The admin logs in through the admin login page. 2. The Admin can access various pages with admin controls. 3. Logs of all the user sessions. 4. User sign-up stats. 5. Logs of all the meetings ever created. 6. Controls for FAQ page. 7. Controls for Quick tour slides. 8. Controls to ban /revoke ban any user. 9. Feature to send an email notification/ platform notification to all users. 10. Controls to edit the email contents. 11. Feature to block domains for new users signing up. 12. Feature to see and ban (take action on, if needed) the reported users.
7	Extensions1(Alt. Flow)	
8	Extensions2(Alt. Flow)	

3) Ban Revoke User use case:

Table 2.3 Ban Revoke User Use Case

1	Use Case	Ban /revoke ban any user
2	Primary Actors	Admin
3	Secondary Actors	UI, System, Database, users
4	Pre-Condition	1. The admin is logged in to the admin dashboard..
5	Post-Condition	1. The admin can ban or revoke any user. 2. The admin can see the ban or revoke history with the details required.
6	Main Success Scenario(Main Flow)	1. The admin clicks on the ban/ revoke page. 2. The admin enters a userID that he wants to ban/revoke ban in the text field. 3. The admin selects a reason-tag from the dropdown button beside it. 4. The admin can add comments in the comments box. 5. The admin clicks on the ban/revoke button. 6. A pop-up with options yes and no will appear. 7. The admin clicks on the yes. 8. A notification appears that the user id XYZ is banned from the platform. 9. The admin can see all the history of the ban/revoke on the bottom half page. 10. These logs can be sorted based on the ban/revoke date, reason-tag.

4) Report the User:

Table 2.4 Report User Use Case

1	Use Case	Report the user
2	Primary Actors	User
3	Secondary Actors	UI, System, Database, secondary user
4	Pre-Condition	1. The user is on the profile page of another user.
5	Post-Condition	1. The user can click on the report the user button if he wishes to report the user. 2. The user can block another user.
6	Main Success Scenario(Main Flow)	1. The user clicks on the profile of the user he wants to report or block. 2. On the profile page, the user can click on the Report or block the user button. 3. Elements - a. Block user b. Spam c. Scam d. Inappropriate e. Harassment f. Hate speech g. Other 4. The user can click any one of them. 5. The user can describe the reason or incident behind the reporting. 6. The user can add image if he wants. 7. The user can click on the checkbox "block the user" if he wants.(Default is already True(already checked)). 8. The user can then click on submit button. 9. The user will be shown a pop up notification/in app notification that the user is blocked or reported.
7	Extensions1(Alt. Flow)	

5) Invite an user:

Table 2.5 Invite User Use Case

1	Use Case	Invite
2	Primary Actors	User
3	Secondary Actors	Friend/ colleague, UI, System, Database
4	Pre-Condition	1. The user is on the dashboard/ homepage.
5	Post-Condition	1. The user is able to invite his friend/ colleague. 2. The friend/ colleague receives an invitation on his email Id. 3. The user receives a confirmation mail.
6	Main Success Scenario(Main Flow)	1. The user clicks on the Invite button on the homepage. 2. A pop-up box appears. 3. With text - " Please, enter the email id of your friend or colleague. " 4. In the textbox available on the pop-up window, the user writes an email id . 5. Another textbox is available for custom invite message , where the user can add his comments for inviting on the platform. 6. The user clicks on the confirm to invite button. 7. If the written email id is a valid email id then the mentioned email id will receive an invitation. 8. The user will be displayed a message - " Invited successfully! "
7	Extensions1(Alt. Flow)	1. If the email ID provided by the user is not in the correct format , then the system will show a message - " Please, enter a valid email ID. "
8	Extensions2(Alt. Flow) Long invite message than the allowed length	1. If the user enters a long comment in the Custom invite Message box, then the system will show a message- " Exceeded maximum length of the invite message. "
9	Extensions3(Alt. Flow) Share with a referral link	1. The user clicks on the Invite button on the homepage. 2. A pop-up box appears. 3. The user can opt to share the link through social media. 4. The user clicks on one of the options - Copy to clipboard, Whatsapp, LinkedIn, FB messenger, etc.

6) Schedule a Video meet use case:

Table 2.6 Schedule a video meet use case

1	Use Case	Schedule a video meet
2	Primary Actors	User1(Mentor), User2(Mentee)
3	Secondary Actors	UI, System, Database
4	Pre-Condition	1. The user is logged in.
5	Post-Condition	1. The user is able to schedule a video meet
6	Main Success Scenario(Main Flow)	1. The user clicks the Create meeting button. 2. A card will expand/ pop up with a search functionality. 3. The user will be shown a search box to search for another professional. 4. From the search result, the user clicks on the professional's name and clicks the "Schedule a meet" button 5. The user will be shown the calendar with other users' available times. (ex. Monday 6pm-9pm) 6. The user selects a time when he wants to connect. 7. The user clicks the Schedule/Confirm Button. 8. Both the users will get notified of the Scheduled video meeting.
7	Extensions1(Alt. Flow)	1. The user is on the Mentor/Mentee's homepage/profile. 2. The user clicks on the Schedule video Meet button. 3. Steps 5 to 8 from the main flow.
8	Extensions2(Alt. Flow)	Confirming from the email.

2.4 TECHNOLOGIES USED

The Frontend of the MentorPro is designed using Next JS and Cascading Style Sheets (CSS) and JavaScript.

The Backend logic was done using Java powered by spring boot framework. For every use case REST APIs were created to hit request and fetch response on the server. Additionally, spring security was used for authentication and role-based authorization on the web application.

Operating System – Windows, Linux and MacOS

Frontend – Next JS, JavaScript and CSS

Backend – Java

Database – MySQL

Framework – Spring Boot

2.4.1 Next JS

Next.js is an open-source web development framework built on top of Node.js enabling React based web applications functionalities such as server-side rendering and generating static websites. React documentation mentions Next.js among "Recommended Toolchains" advising it to developers as a solution when "Building a server-rendered website with Node.js". Where traditional React apps can only render their content in the client-side browser, Next.js extends this functionality to include applications rendered on the server side.

Following are the benefits of using Next:

- Adaptability and responsiveness
- Data security
- Faster time to market
- Fully Omni channel
- Short page load time
- Support on demand

2.4.2 Spring Boot

Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications. Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run. You can get started with minimum configurations without the need for an entire spring configuration setup.

Spring Boot automatically configures your application based on the dependencies you have added to the project by using `@EnableAutoConfiguration` annotation. For example, if MySQL database is on your class path, but you have not configured any database connection, then Spring Boot auto-configures an in-memory database.

Spring Boot provides the facility to enable admin-related features for the application. It is used to access and manage applications remotely. We can enable it in the Spring Boot application by using **`spring.application.admin.enabled`** property.

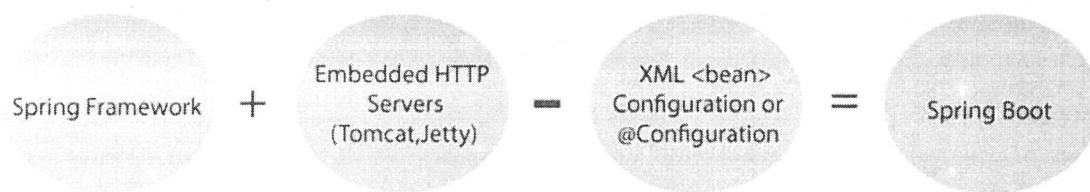


Figure 2.2 Spring Boot Framework

Following are the features of using spring boot:

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

2.4.3 Spring Security

Spring Security is a framework provided by spring that helps to customize an access and authentication process. It plays a very crucial role in terms of securing the applications.

Spring Security, mainly focuses on authentication and authorization to provide all benefits to java applications. It is very helpful and provides an easy approach to apply in real projects. And, permits to do custom modifications in real projects.

Following are the features of the spring security:

- Supports authentication and authorization processes comprehensively and extensively.
- Prevents from cross-site forgery, session fixation, clickjacking, etc.
- Integrates the Servlet API.
- Provides integration with Spring Web MVC optionally

Flow of Spring Security is as follows:

- **SecurityFilterChain** : Spring Security maintains a filter chain internally where each of the filters is invoked in a specific order. Each filter will try to process the request and retrieve authentication information from it. For example, we have the UsernamePasswordAuthenticationFilter which is used in case of a POST request with username and password parameters
- **AuthenticationManager**: This is an interface whose implementation (ProviderManager) has a list of configured AuthenticationProviders that are used for authenticating user requests.

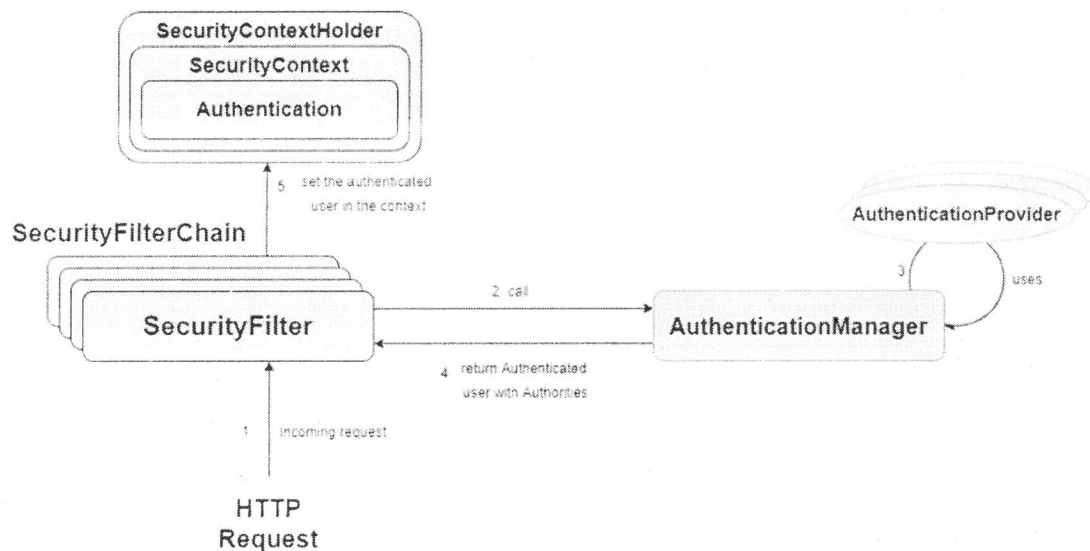


Figure 2.3 User Credential Journey

- **AuthenticationProvider** : An AuthenticationProvider is an abstraction for fetching user information from a specific repository (like a database, LDAP, custom third party source, etc.). It uses the fetched user information to validate the supplied credentials.
- When the Authentication is successful, and a matching user is found, an Authentication Object containing the user Authorities (which will be used to manage the user access to the system's resources) is returned and set into the **SecurityContext**.

2.4.4 Docker

Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multi cloud environments.

Developers can create containers without Docker, but the platform makes it easier, simpler, and safer to build, deploy and manage containers. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API.

Following are the features of using Docker:

- Faster and easier configuration.
- Application isolation.
- Increase in productivity.

- Swarm.
- Services.
- Routing Mesh.
- Security Management.
- Rapid scaling of Systems

2.4.5 GitHub

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration. GitHub's interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line. Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

Following are the feature of suing Github:

- Collaboration
- Integrated issue and bug tracking
- Graphical representation of branches
- Git repositories hosting
- Project management
- Team management
- Code hosting
- Track and assign tasks

2.4.6 GitHub Actions

GitHub Actions is a CI/CD platform for automating your build, test, and deployment pipelines. You may set up workflows to build and test each pull request in your repository, as well as publish merged pull requests to production.

GitHub Actions goes beyond DevOps by allowing you to perform processes in response to other events in your repository. For example, anytime someone files a new issue in your repository, you may execute a workflow to automatically add the required labels.

You can execute your workflows on GitHub's virtual machines running Linux, Windows, and macOS, or you can host your own self-hosted runners in your own data centre or cloud infrastructure.

Components of Github Actions are:

1) Workflow:

A workflow is a configurable automated process that will run one or more jobs. Workflows are defined by a YAML file checked in to your repository and will run when triggered by an event in your repository, or they can be triggered manually, or at a defined schedule.

2) Events:

An event is a specific activity in a repository that triggers a workflow run. For example, activity can originate from GitHub when someone creates a pull request, opens an issue, or pushes a commit to a repository. You can also trigger a workflow run on a schedule, by posting to a REST API, or manually.

3) Jobs:

Job is a set of steps in a workflow that execute on the same runner. Each step is either a shell script that will be executed, or an action that will be run. Steps are executed in order and are dependent on each other. Since each step is executed on the same runner, you can share data from one step to another.

4) Actions:

An action is a custom application for the GitHub Actions platform that performs a complex but frequently repeated task. Use an action to help reduce the amount of repetitive code that you write in your workflow files. An action can pull your git repository from GitHub, set up the correct toolchain for your build environment, or set up the authentication to your cloud provider.

2.4.7 Database Migrations

Database migration is the process of migrating data from one or more source databases to one or more target databases by using a database migration service. When a migration is finished, the dataset in the source databases resides fully, though possibly restructured, in the target databases.

1) Flyway:

Flyway is an open-source programme for automating and version-based database migrations that is distributed under the Apache License 2.0. It allows you to write SQL scripts or Java code to describe the required update operations.



Simple to use

Flyway lets you regain control of your migrations with ease.



Solves one problem well

Flyway migrates your database, so you don't have to worry about it.



Made for CI/CD

Releases have never been this easy.

Figure 2.4 Flyway

The migration can then be started manually, automatically as part of your build process, or integrated into your Java application. The benefit of this method is that Flyway automatically detects and performs the appropriate update activities. As a result, you

won't need to know which SQL update commands to run to update your current database. To migrate the database from one version to the next, you and your coworkers just describe the update actions. Flyway also identifies the current version and updates it as needed.

2.4.8 Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff. The main objective of unit testing is to isolate written code to test and determine if it works as intended.

Unit testing is an important step in the development process, because if done correctly, it can help detect early flaws in code which may be more difficult to find in later testing stages.

Following are the features of unit testing:

- The earlier a problem is identified, the fewer compound errors occur.
- Costs of fixing a problem early can quickly outweigh the cost of fixing it later.
- Debugging processes are made easier.
- Developers can quickly make changes to the code base.
- Developers can also re-use code, migrating it to new projects.

CHAPTER 3: SOFTWARE DESIGN

3.1 DESIGN STRATEGY

Software design is the process of developing software to solve one or more problems. It converts user and client requirements into a format that programmers can use for development and implementation.

Problem addressing and software solution planning are common tasks in software design. This encompasses component and algorithm design at the low level, as well as high-level architecture design.

Furthermore, software design shifts the focus from the problem domain to the solution drama. It seeks to describe how to meet the requirements in Software Requirement Specifications (SRS).

Software Design Strategy is a discipline that assists businesses in determining what to build, why to do it, and how to innovate contextually in the short and long term. This method entails strategic design as well as the interaction between design and business strategy.

The software design strategy is primarily concerned with coordinating design activities as they occur.

3.1.1 Function oriented design

The system in a function-oriented architecture is made up of numerous smaller sub-systems called functions. These functions are capable of completing important system tasks. All functionalities are viewed from the top of the system.

The system is developed according to the functionality it provides in a function-oriented design. The system is initially inspected broadly, and then each of its functions is examined to determine which sub-functions are responsible for carrying out that functionality.

There is a centralised function that consists of various functions in this type of architecture. The programme follows the same pattern.

The system in a function-oriented architecture is made up of numerous smaller sub-systems called functions. These functions are capable of completing important system tasks. All functionalities are viewed from the top of the system.

The system is developed according to the functionality it provides in a function-oriented design. The system is initially inspected broadly, and then each of its functions is examined to determine which sub-functions are responsible for carrying out that functionality.

There is a centralised function that consists of various functions in this type of architecture. The programme follows the same pattern.

3.2 FEASIBILITY STUDY

3.2.1 Technical Feasibility

A technical feasibility study looks into the specifics of how a system or solution will be built. Technical analysis assesses the system's technical qualities while also gathering extra data on performance, dependability, maintainability, and productivity. The feasibility study of the project is examined in this part.

Essentially, the goal of this research is to ensure that the suggested system software does not cause the user any harm or inconvenience. It determines whether the suggested project is feasible. This research necessitates a thorough understanding of the system.

3.2.2 Time Schedule Feasibility

The Project has simple working and basic requirement can be satisfied with the allotted time period.

3.2.3 Operational Feasibility

The proposed system will improve all users' operational efficiency. As a result, the system's throughput and time savings are as expected.

This system will ensure and meet the needs defined during the system's requirement analysis phase.

For the same definition, the API will be user-friendly and work with any frontend system, making it extremely helpful and simple to use.

3.2.4 Implementation Feasibility

Users can use Postman or any other API testing tools to directly access APIs, or they can leverage APIs to develop frontends.

Customers will be able to engage any of the available pros when the implementation is complete.

CHAPTER 4: CONSTRUCTION

4.1 GENERAL INTRODUCTION

Software testing is the process of verifying the accuracy of software by considering all aspects and evaluating the implementation of software components in order to detect bugs, mistakes, or flaws. Readability, scalability, portability, reusability, and usability are all software attributes.

Software testing provides a stand-alone view and objectivity of the software while also ensuring its correctness. It entails putting all parts through the appropriate procedures to ensure that they meet the set requirements. The method also provides information about the product's nature to the buyer.

The car portal is tested using the manual testing procedure. Manual testing is the process of testing software without the use of any automated tools. Manual testing is divided into three categories: white box testing, black box testing, and grey box testing. Manual testing is carried out utilising the functional unit testing technique, which is a subset of black box testing.

Functional testing focuses on ensuring that the software's functionalities are validated by ensuring that it meets the requirements. Each feature is evaluated by providing input, calculating the result, and comparing the actual output to the predicted output.

4.2 TEST SCENARIO

4.2.1 Login

- a) Test case 1: Valid email and password
- b) Test case 2: Invalid email and password
- c) Test case 3: Empty email and password field

Table 4.1 Login Test

Test Case Number	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
TC1	Check user login with valid credentials	Email: mehulkhandelwal1290@gmail.com Password: "mentorpro"	User should login into portal	As expected	Pass
TC2	Check user login with invalid credentials	Email: mehulkhandelwal1290@gmail.com Password: mentor	User should not login into portal	As expected	Pass
TC3	Check user login with empty credentials	Email: Password:	User should not login into portal	As expected	Pass

4.2.2 Block Domain

- a) Test case 1: Input validation with invalid input
- b) Test case 2: Empty field
- c) Test case 3: Subordinate

Table 4.2 Block Domain Test

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation on domain information with invalid input	Point out the error	As expected	Pass
TC2	Check domain information with empty fields	Point out the No element found field	As expected	Pass
TC3	Check for existing domain	Exception should be thrown. Domian already Exists	As expected	Pass

4.2.3 Invite User

- a) Test case 1: Input validation with invalid input
- b) Test case 2: Empty field
- c) Test case 3: Subordinate

Table 4.3 Invite User Test

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check for empty referral email fields	Point out the empty filed	As expected	Pass
TC2	Check invalid information of referred email user	Point out the error in mailbox	As expected	Pass
TC3	Check for invalid refferals to ID	Exception should be thrown	As expected	Pass

4.2.4 Signup

- a) Test case 1: Valid email and password
- b) Test case 2: Invalid email and password
- c) Test case 3: Empty email and password field

Table 4.4 Signup Test

Test Case Number	Test Case Description	Test Data	Expected Result	Actual Result	Pass/ Fail
TC1	Check user signup with valid credentials	Email: mehulkhandelwal1290@gmail.com Password: "mentorpro"	User should receive verification mail	As expected	Pass
TC2	Check user signup with invalid credentials	Email: mehulkhandelwal1290gmail.com Password: "mentor"	User should not receive verification mail	As expected	Pass
TC3	Check user singup with empty credentials	Email: Password:	Singup Details Not Found Exceptions	As expected	Pass

4.2.5 Report User

- d) Test case 1: Input validation with invalid input
- e) Test case 2: Empty field
- f) Test case 3: Subordinate

Table 4.5 Report user Test

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check input validation on report information with invalid input	Point out the error	As expected	Pass
TC2	Check report reason information with empty fields	Point out the No element found field	As expected	Pass
TC3	Check for reportings	Exception should be thrown. User already reportes	As expected	Pass

4.2.6 Ban/Revoke User

- d) Test case 1: Input validation with invalid input
- e) Test case 2: Empty field
- f) Test case 3: Subordinate

Table 4.6 Ban Revoke User

Test Case Number	Test Case Description	Expected Result	Actual Result	Pass/ Fail
TC1	Check for empty user ban fields	Point out the empty filed	As expected	Pass
TC2	Check invalid information of banning user	Point out the error in mailbox	As expected	Pass
TC3	Check for already banned users	Exception should be thrown. UserAlreadyBanned Exception.	As expected	Pass

CHAPTER 5: CONCLUSION

This Project aims at solving the problem of some traditional systems where it was difficult to handle the mentee records and give timely feedback to the mentees. For the users of an organization, who need assistance regarding any technical and non-technical queries, this system provides online communication with mentors in an easy and effective manner which saves time and energy.

During this project I have accomplished all the objectives and this project meets the needs of the lending organization. At ZopSmart Technologies, as an intern, it was a great learning experience, full of knowledge and excitements. I am truly grateful for the wonderful opportunity.

REFERENCES:

- Baeldung. 2022. Java, spring and Web Development tutorials. [online] Available at: <<https://www.baeldung.com/>> [Accessed 12 May 2022].
- www.javatpoint.com 2022. Tutorials - Javatpoint. [online] Available at: <<https://www.javatpoint.com/>> [Accessed 12 May 2022].
- Docker. 2022. Home - Docker. [online] Available at: <<https://www.docker.com/>> [Accessed 12 May 2022].
- GeeksforGeeks. 2022. GeeksforGeeks | A computer science portal for geeks. [online] Available at: <<https://www.geeksforgeeks.org/>> [Accessed 12 May 2022].
- Stack Overflow. 2022. Stack Overflow - Where Developers Learn, Share, & Build Careers. [online] Available at: <<https://stackoverflow.com/>> [Accessed 12 May 2022].

INTERNSHIP CERTIFICATE

Date: 06/05/2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mehul Khandelwal has done his internship in software development at ZopSmart Technology Pvt Ltd., Bengaluru, from 03/01/2022 to 06/05/2022 for the duration of 4 months.

He has worked on a project titled Mentor Pro. This project was aimed at providing one stop portal for professional guidance and mentorship. As part of the project, he has worked in implementing necessary REST APIs, database migrations, authorizations, and other necessary duties for shaping the project.

During his internship he has demonstrated his skills with self-motivation to learn new skills. His performance exceeded our expectations, and he was able to complete the project on time.

We wish him all the best for his upcoming career.

ZopSmart Technology Pvt Ltd.



Manager- Human Resources

