

Asset Management Portal

Computer Science and Engineering

By

Harshit Chanana 181256

Under the supervision of

Indrayani Sharad

Project Lead (Infosys)



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology, Wagnaghat,
173234, Himachal Pradesh, INDIA**

DECLARATION

I on my behalf declare that the project that I have done has been made by us under the direction of (Dr Indrayani Sharad) the Jaypee University of Information Technology. I also announce that neither this project nor any part of this project has been submitted elsewhere for any purpose.

Supervised to:
Indrayani Sharad
Project Lead(Infosys)

Submitted by:
Harshit Chanana 181256
Computer Science & Engineering Department Jaypee University of Information
Technology

CERTIFICATE

This is to certify that the project which is being represented in the project report named “School data management” in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and represented to the Department of Computer Science and Engineering (CSE), Jaypee University of Information Technology, Waknaghat is an authentic record of work done by Harshit Chanana (181256) during the period from January 2022 to May 2022 under the supervision of Dr Monica Bharti, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Harshit Chanana 181256

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

Indrayani Sharad Project Lead (Infosys)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat

ACKNOWLEDGEMENT

Firstly, I am so gratefulness to Almighty God and thankful for his divine grace that made it possible to successfully complete the project work in a smooth way.

I am grateful and wish my profound indebtedness to my supervisor Dr. Indrayani Sharad, who is in department of computer science Jaypee University of Information Technology, Waznaghat. where she was very helpful and guided us throughout the thesis to carry out this project and also provide us with the necessary materials that we need. Her endless scholarly guidance, patience, continual encouragement, constant scholarly guidance and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude and thankful to Dr. Indrayani Sharad, Department of computer science, for her considerate nature and kind help to finish my project on time.

I would also generously thank each of the individuals who have helped me continuously and straightforwardly or in a roundabout way to make this project a success. In this odd and strange situation, I also want to thank the various staff individuals, both educating and non-educating, which have developed their convenient help and facilitated my undertaking.

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

Harshit Chanana 181256

TABLE OF CONTENT

<i>Content</i>	<i>Page Number</i>
1.Chapter 1 (Introduction)	1-9
2.Chapter 2 (Literature Survey)	10
3.Chapter 3 (System Development)	11-22
4.Chapter 4 (Performance Analysis)	23-26
5. Chapter 5 (Conclusions)	26-33
References	34

LIST OF FIGURES

Fig No.	Figure Name	Page No.
1.1	Database	3
1.2	Spring Boot	6
3.1	Agile methodology	12
3.2	Data Flow Diagram	13
3.3	MVC	14
3.4	View	15
3.5	Models Used	16-21
3.6	Login design	22
4.1	Spring Initializr	26

ABSTRACT

Asset Management system is used to maintain assets of company requires a significant amount of time if we do it without software. Every organization has their own Asset Management System in order to perform the resource activities. Managing Asset or utilizing assets or keeping track of assets are significant task of IT team. The main work of the IT team is to maintain all assets of the company like laptop and any other devices and also maintain the condition of that Asset. In order to support the IT's there are some electronic based systems called AMS- Asset management system. Organizations maintained assets with excel sheets which is a very tedious job. But this application is cost effective one that allows them to manage their asset's data in a simple manner. This project belongs to a category of web application that can be accessed through PC with internet connection. This Asset Management System allows the IT team to provide various types of permission to other IT members and also manage Asset's data, status, category, location, etc.

In this project I have used Spring Boot and Angular in order to save information in database and utilize the data in such a way which is fast and effective and display the status of Asset of the organization. This project contains basic CRUD functions and even some complex function which will be helpful in order to find the data of a particular user and asset and management of assets .

CHAPTER 01

INTRODUCTION

- 1.1 Introduction**
- 1.2 Problem Statement**
- 1.3 Objective**
- 1.4 Methodology**
- 1.5 Organization**

1.1 INTRODUCTION

This project is intended to replace the Manually managed system and provide a platform that is more efficient, reliable and robust. This project is based on Angular along with Spring Boot and MySQL as database. MySQL is a type of database which instead of using typical table use it uses fields and document which help the data to make it more scalable and useful . Spring Boot act as a back door entry to data in ease we can do all the CRUD operations on data in microseconds due to the features provided by the spring boot. Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications

Asset Management System is an application used by the company to automate and keep track of the equipment and inventory that are vital to the day-to-day operation of their businesses. Exactly how those assets are managed, though, is highly variable. A lot of organizations are managing their equipment and inventory through a manual process, including spreadsheets over which an employee or employees are tasked with the responsibility of maintaining.

1.2 PROBLEM STATEMENT

IT asset management ties the assets with the IT infrastructure of the organization. With a robust asset management system, management and IT professionals can review and monitor all types of assets within the organization. The information can be used to make detailed decisions about the purchase and other aspects of the asset's life cycle. So, the purpose of IT asset management is to:-
Effectively help manage the assets.

- Improve visibility of assets.
- Ensure optimum utilization of assets.
- Reduce IT and software costs.
- Ensure compliance with regulatory requirements.

So basically Asset Management portal is an approach towards the monitoring of all types of assets within the organization. This will help organization in making decisions about all the aspects of assets.

1.3 OBJECTIVE

The objective of the IT asset management system is to provide an organization with deep knowledge of its information systems to use this information for the identification and rapid resolution of problems. And also complete information about the configuration of assets and their relationship among each other and also make sure that this data is available whenever it is needed.

The goal of this project is to reduce the effort required to maintain a variety of assets and automate the task of managing assets. Using the software can help in efficient resource planning.

All in all the main aim to make this application is to make the best use of technology in order to do CRUD operation on the data and make it free and easy to use I have used different technology like maven, swagger to make the app more interactive. This project can be used where the data is in abundance like schools, institutes etc.

1.4 METHODOLOGY

1.4.1 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language, programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability.

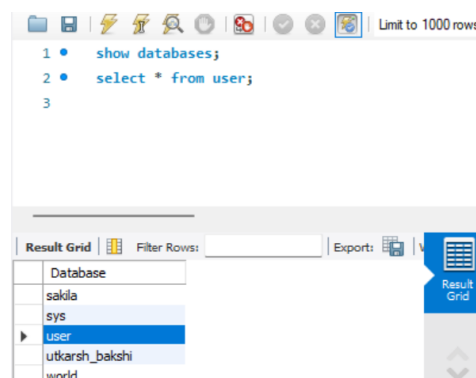


Fig 1.1 Database

1.4.2 Backend

1.4.2.1 Spring Boot

Prior to the introduction of Enterprise Java Beans (EJB), Java developers had to use JavaBeans to build web applications. Although JavaBeans contributed to the development of user interface (UI) components, they were unable to provide the services, such as performance management and security, which were required to develop robust and secure business applications. The arrival of EJB has been identified as a solution to this problem. EJB expands Java components, such as the Web and business components, and provides services that assist in the development of business applications. However, building a business plan with EJB was not easy, as the developer needed to perform a variety of tasks, such as doing Home and Remote interfaces and using the life-cycle repetition methods that lead to EJB code delivery problems. Looking for an easy way to improve business applications.

The Spring Framework has emerged as a solution to all of these problems. This framework uses various new strategies such as Aspect-Oriented Programming (AOP), Plain Old Java Object (POJO), and dependency injection (DI), to improve business plans, thus eliminating them. The complexities involved in building business applications using EJB, Spring is an open source lightweight solution that allows Java EE 7 developers to build simple, reliable, and efficient business applications. This framework focuses on providing a variety of ways to help you manage your business assets. It has made the development of Web applications much easier compared to older Java frameworks and Application Programs (APIs), such as Java website integration (JDBC), JavaServer Pages (JSP), and Java Servlet.

The Spring Framework can be considered as a subgroup of structures, also called layers, such as Spring AOP, Spring Object-Relational Mapping (Spring ORM), Spring Web Flow, and Spring Web MVC. You can use any of these modules separately while building a Web application. Modules can also be grouped together to provide the best performance on a Web application.

Features of Spring Framework

The features of the Spring framework such as IoC, AOP, and performance management, make it unique among the framework frameworks. Some of the most important features of the Spring Framework are the following:

IoC Container:

Refers to the main container that uses the DI or IoC pattern to explicitly provide object reference in class during operation. This pattern serves as another pattern for the service area. The IoC container contains an integral code that handles application configuration management.

The Spring Framework offers two packages, namely `org.springframework.beans` and `org.springframework.context` which assists in providing the functionality of the IoC container.

Data access framework:

Allows developers to use continuous APIs, such as JDBC and Hibernate, to store persistent data on a website. It helps solve various engineer problems, such as how to communicate with a website connection, how to make sure communications are closed, how to deal with differences, and how to implement transaction management. It also enables developers to write easily access code for persistence data throughout the application.

Spring MVC Framework:

Allows you to create web applications based on MVC architecture. All user-made requests first go through the controller and are sent to a different view, that is, to different JSP pages or Servlets. Form management and verified form features of the Spring MVC framework can be easily integrated with all popular viewing technologies such as JSP, Jasper Report, FreeMarker, and Velocity.

Performance Management:

Helps in managing application performance without touching its code. This framework provides the Java Transaction API (JTA) for global applications hosted by application server and local operations managed using JDBC Hibernate, Java Data Objects (JDO), or other data access APIs. It allows the developer to model multiple tasks on the basis of advertising and programmatic management.

Spring Web Service:

Produces final web service points and definitions based on Java classes, but is difficult to manage in the application. To solve this problem, the Spring Web Service provides horizontally controlled alternatives to Extensible Markup Language (XML) (XML learning and deception) strategy. Spring provides an effective map to transfer XML message request to an object and the developer can easily distribute an XML (object) message between two machines.

JDBC Abbreviation Layout:

Helps users manage errors easily and effectively. JDBC editing code can be minimized if this removal layer is applied to a web application. This layer handles differently as `DriverNotFound`. All `SQLExceptions` are translated into the `DataAccessException` class. Spring data access exceptions are not specific to JDBC and that is why Data Access Equipment (DAO) is not limited to JDBC.

Spring TestContext Frame:

Provides unit resources and integration testing for Spring applications. In addition, the Spring TestContext framework provides direct integration test functions such as context management and DI caching for test fixtures, as well as practical test management with automatic recurring semantics.

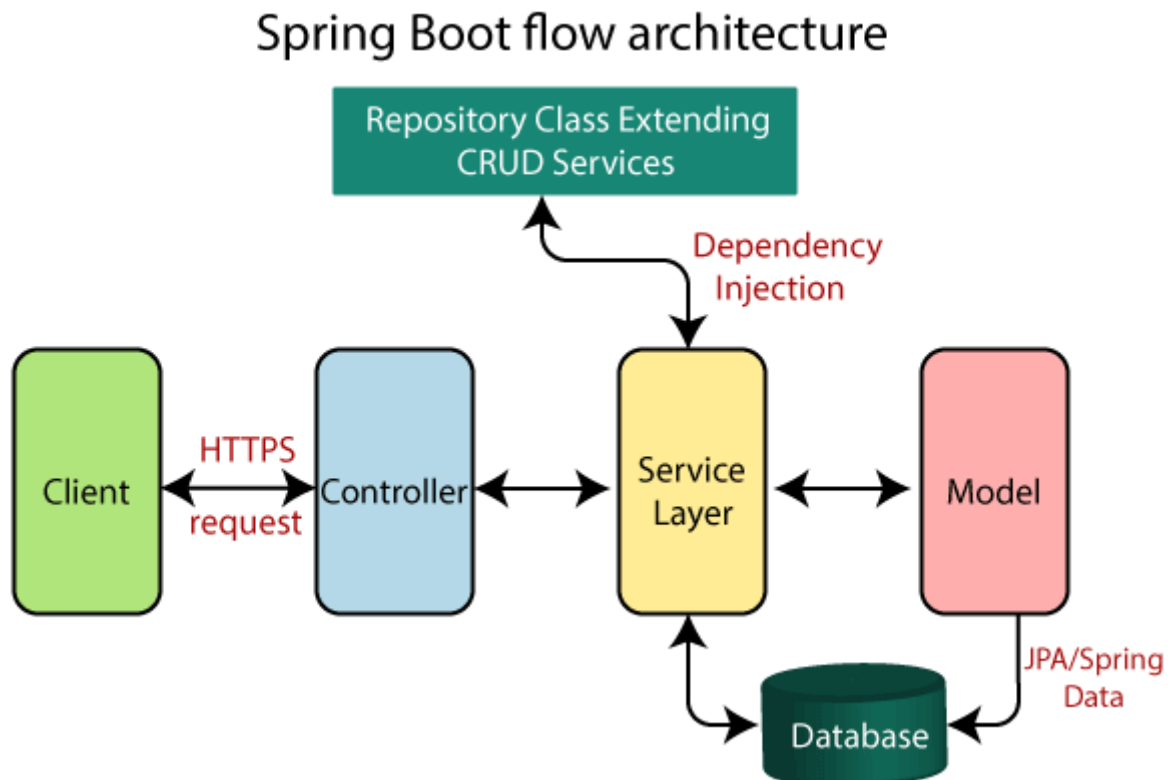


Fig 1.2 Spring Boot

1.4.2.2 Autoconfiguration

Autoconfiguration means that applications are initialized with pre- set dependencies that you don't have to configure manually. As Java Spring Boot comes with built-in autoconfiguration capabilities, it automatically configures both the underlying Spring Framework and third-party packages based on your settings (and based on best practices, which helps avoid errors).

Opinionated approach

Spring Boot uses an opinionated approach to adding and configuring starter dependencies, based on the needs of your project. Following its own judgment, Spring Boot chooses which packages to install and which default values to use, rather than requiring you to make all those decisions yourself and set up everything manually.

Standalone applications

Spring Boot helps developers create applications that *just run*. Specifically, it lets you create standalone applications that run on their own, without relying on an external web server, by embedding a web server such as Tomcat or Netty into your app during the initialization process. As a result, you can launch your application on any platform by simply hitting the Run command.

1.4.3 Frontend

HTML

The Hyper-Text Mark-up Language or HTML is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, which reduces complexity and repetition in the structural content; and enable the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across

web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

Angular Framework

Angular is a TypeScript-based free and open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS.

Angular is used as the frontend of the MEAN stack, consisting of MySQL database, Express.js web application server framework, Angular itself (or AngularJS), and Node.js server runtime environment. Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into NgModules. NgModules collect related code into functional sets; an Angular application is defined by a set of NgModules. An application always has at least a root module that enables bootstrapping, and typically has many more feature modules.

Components define views, which are sets of screen elements that Angular can choose among and modify according to your program logic and data. Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.

1.5 ORGANIZATION

Chapter 1 contains the Introduction, Problem Statement, Objective, and Methodology of the Project or System.

Chapter 2 contains the literature survey in which some of the previous works is studied and compiled in order to make this model.

Chapter 3 discusses the system development as it contains all the computational, mathematical, analytical, theoretical data.

Chapter 4 is all about the performance analysis in which different models' results are compared, with algorithm improvement and outputs

Chapter 5 discusses the conclusions, future scope of the model, its applications in real world, advantages and limitations.

Then I have the References section.

Last is the Appendices section which includes the source code of model.

CHAPTER 02

LITERATURE SURVEY

Companies manage their equipment and inventory through a manual process, including spreadsheets over which an employee or employees are tasked with the responsibility of maintaining.

Spring Boot makes it easy to create standalone applications, with a production level that you can "just use".

We take a visual view of the spring stadium and the libraries of foreign companies to get started with a little controversy. Most Spring Boot apps require minimal Spring configuration.

If you are looking for information on a particular version, or instructions on how to improve from a previous release, check out the project release notes section in our wiki.

Features

Create independent spring apps

Embed Tomcat, Jetty or Undertow directly (no need to send WAR files)

Provide a 'first' reliance on ideas to make your construction preparation easier

Automatically customize Spring and foreign libraries whenever possible

Provide ready-to-produce features such as metrics, health tests, and external configurations

There is no code production at all and there is no need for XML configuration

Due to manual processes, IT supervisors suffer problems like forgotten items, broken or lost equipment, human error, hours spent searching for items that have mysteriously disappeared, inaccurate information needed for accounting and compliance, customer dissatisfaction. So with the use of Angular framework and Spring Boot we have executed the project.

CHAPTER 03

SYSTEM DEVELOPMENT

- 3.1 Requirement Specifications**
- 3.2 Design and Planning**
- 3.3 Features used**
- 3.4 Models used**
- 3.5 Computational development**

3.1 Requirement Specifications

Hardware Requirements

Processor	1 GHz
RAM	1 GB
Hard Disk Space	16 GB
Display	800x600

Software Requirements

Front-end	HTML, CSS, BOOTSTRAP, ANGULAR
Back-end	JAVA SPRINGBOOT
Tools	VS Code, Postman, Json
Database	MySQL

3.2 Design and Planning

Agile Model

Agile model is the combination of iterative and incremental process models. The steps involved in agile SDLC models are:

- Requirement gathering
- Requirement Analysis
- Design
- Coding
- Unit testing
- Acceptance testing

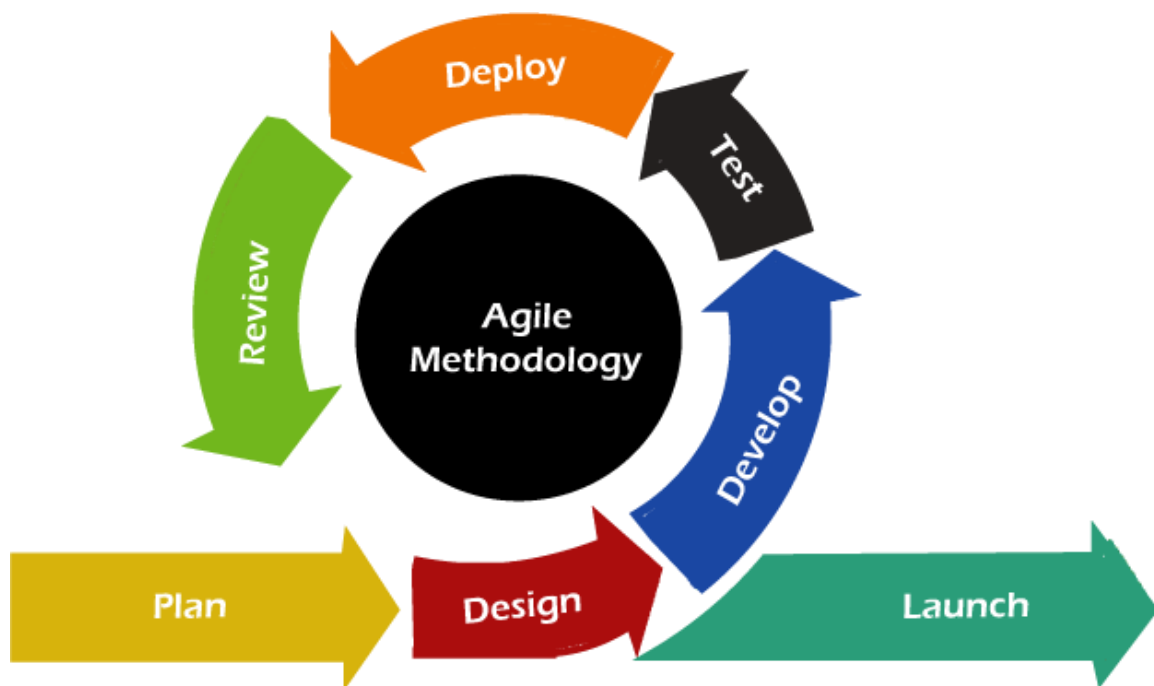


Fig 3.1 Agile Methodology

Data Flow Diagram

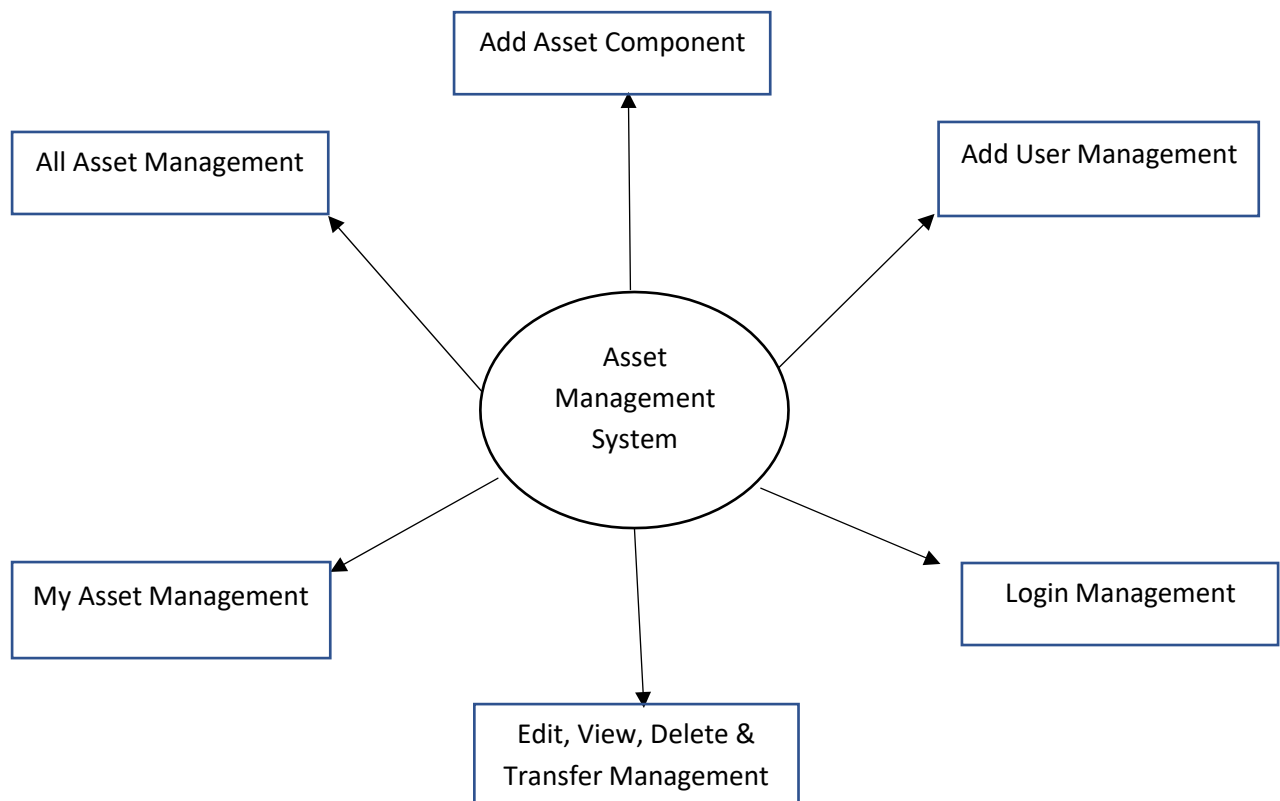


Fig 3.2 Data Flow Diagram

3.3 Features Used **Spring MVC:**

Spring MVC is a Java framework used to build web applications. It follows the Model-View-Controller design pattern. It uses all the basic features of the main spring frame such as control modification, dependence injection.

Spring MVC provides a great solution to use MVC in the spring framework with the help of the DispatcherServlet. Here, DispatcherServlet is a class that receives an incoming application and places it on a map that uses the appropriate as controls, models, and views.

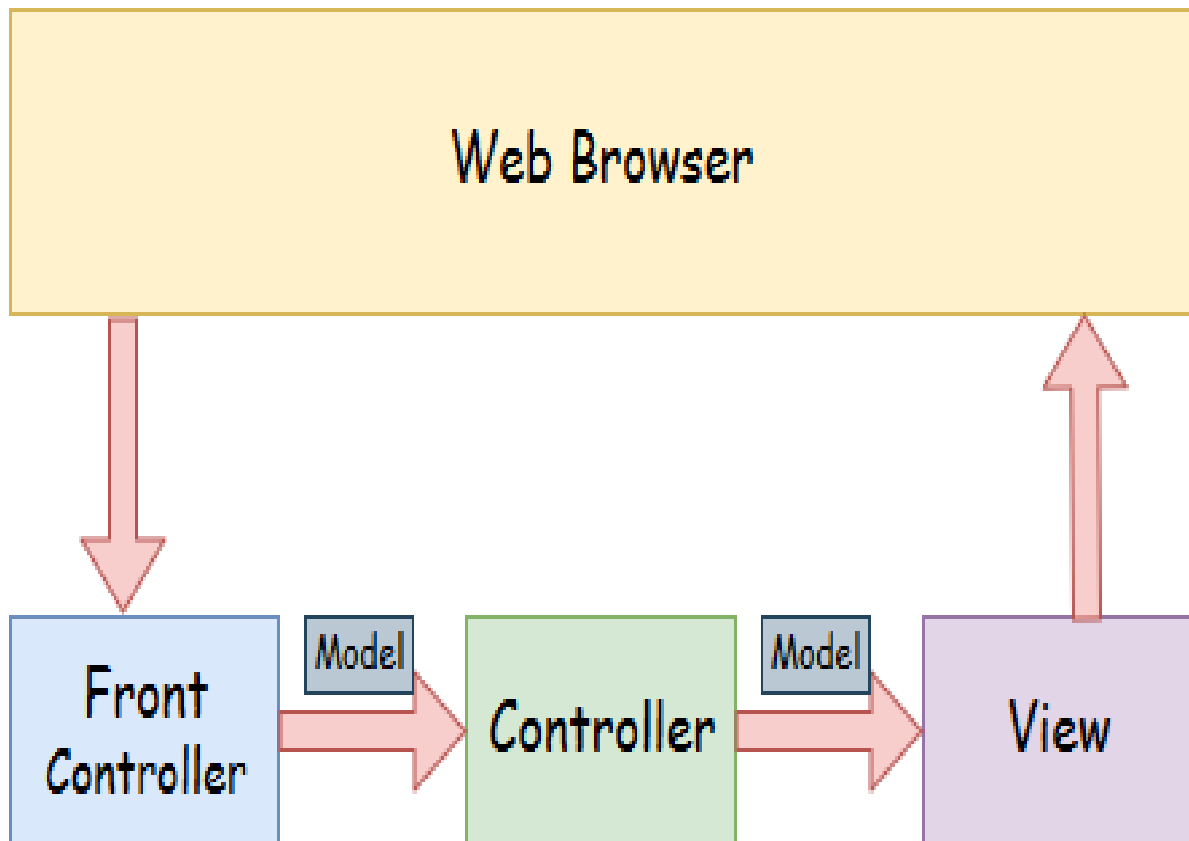


Fig 3.3 MVC

Model - Model contains application data. Data can be a single object or a set of objects.

Controller - The controller contains the business intelligence of the application. Here, the @ Controller annotation is used to mark a category as a controller.

View - View represents information provided in a specific format. Typically, JSP + JSTL is used to create a web page. Although spring also supports other viewing technologies such as Apache Velocity, Thymeleaf and FreeMarker.

Front Control - In Spring Web MVC, the DispatcherServlet section acts as the front controller. There is a responsibility to manage the flow of the Spring MVC application.

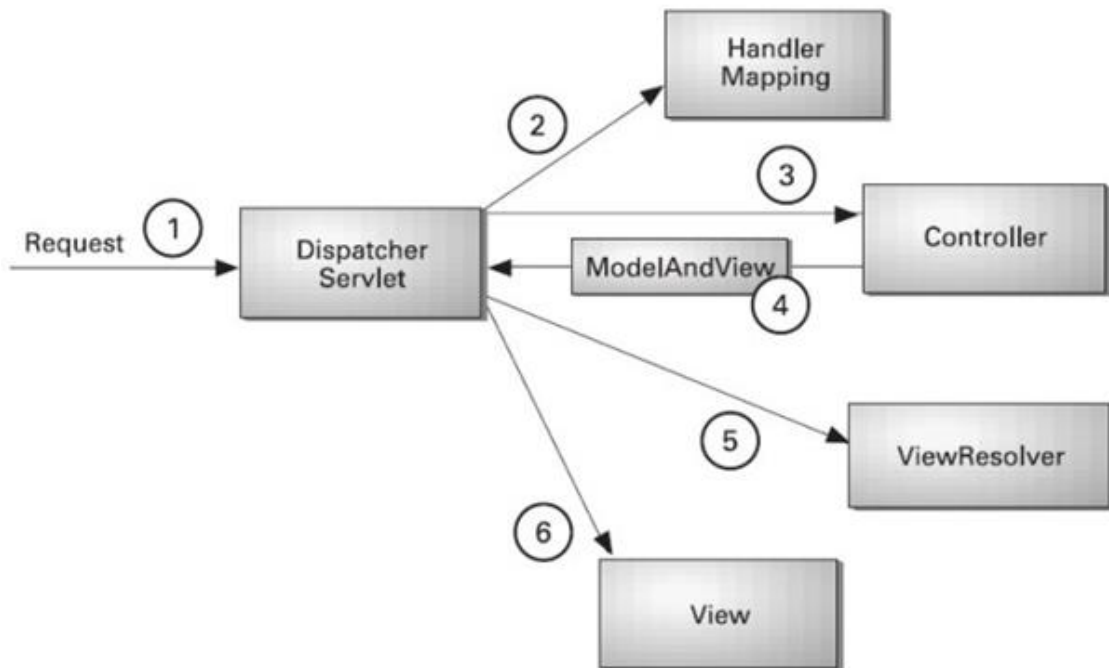


Fig 3.4 View

As shown in the picture, every incoming request is received by the DispatcherServlet which serves as the front controller.

DispatcherServlet retrieves the map holder from the XML file and transfers the application to the controller.

The controller returns the ModelAndView object.

DispatcherServlet checks the installation of the viewing solution in the XML file and asks for the specified viewing component.

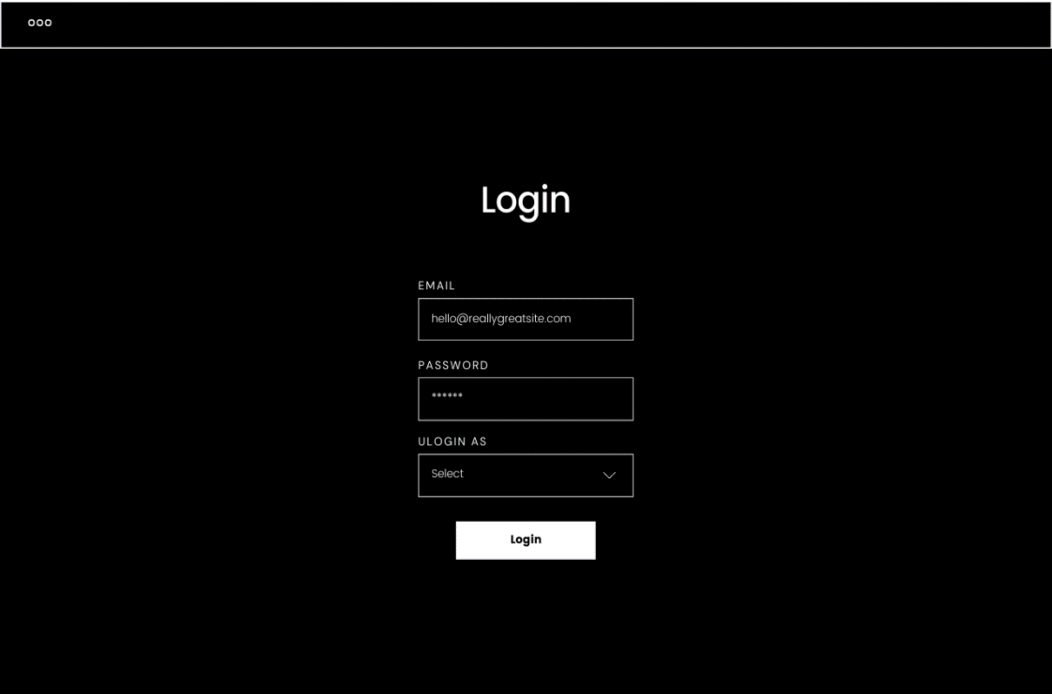
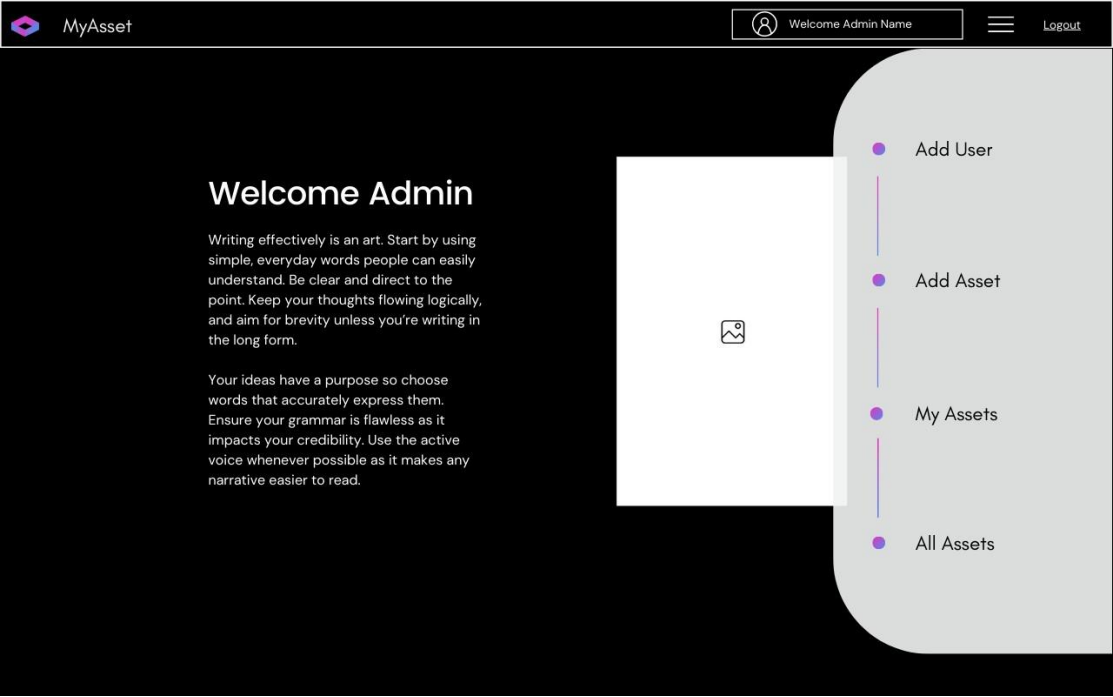
Postman

APIs make it possible for any two separate applications to transfer and share data between them. They also make it easier for an application's users to execute actions without having to use the application's GUI. From the developers' POV, it's an easy way to execute certain functionalities of their app and test it as well.

Using APIs on a daily basis might become cumbersome, as one might have dozens or even hundreds of APIs that he or she needs to use or test. That makes it difficult to keep up with their exact request's address(es), header(s), authorization credential(s) etc., and by that make it harder to test the API for functionality, security and exception handling.

Postman is an API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. The result - more efficient and less tedious work

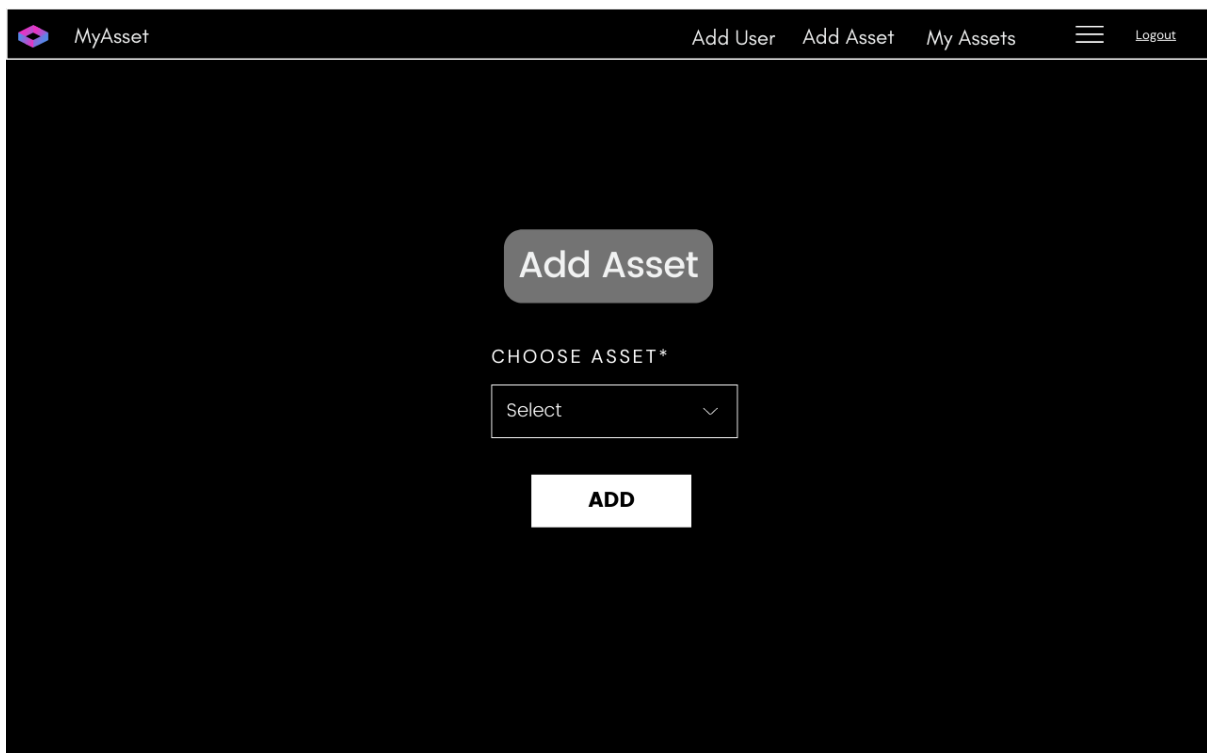
3.4 Models description



This is the welcome page of our portal which will accept the login using Infosys domain.

Add Asset:

- Add Asset screen will contain a dropdown option on screen with 6 asset options to choose from. As soon as admin selects the asset, respective form will pop-up to fill the required details.
- The dropdown will have following options- Iphone, Ipad, Android, iMac, Macbook And Macmini in alphabetical order.
- The form will contain mandatory and non- mandatory fields and some additional fields for some selected asset like Imac, Macbook and Macmini
- Ipone and Ipad will have the same form with same mandatory and non-mandatory fields
- Android will have different form with different fields with no Udidfield. Macbook, Macmini and Imac will have the same fields and same mandatory and non-mandatory fields
- A dialog box on successful submission of Add Asset Form & it will Re-Route user to respective home page and refreshed the page on click



The screenshot shows the 'Add Asset' form in the 'MyAsset' portal. The header bar is dark blue with the 'MyAsset' logo on the left and navigation links 'Add User', 'Add Asset', 'My Assets', and 'Logout' on the right. The main content area is white. At the top center is a large blue button labeled 'Add Asset'. Below it is the text 'CHOOSE ASSET*' followed by a dropdown menu with 'Select' and a downward arrow. At the bottom center is a red button labeled 'ADD'.

My Asset:

My Asset page will have a table on the screen where all assets possessed by the admin will be listed.

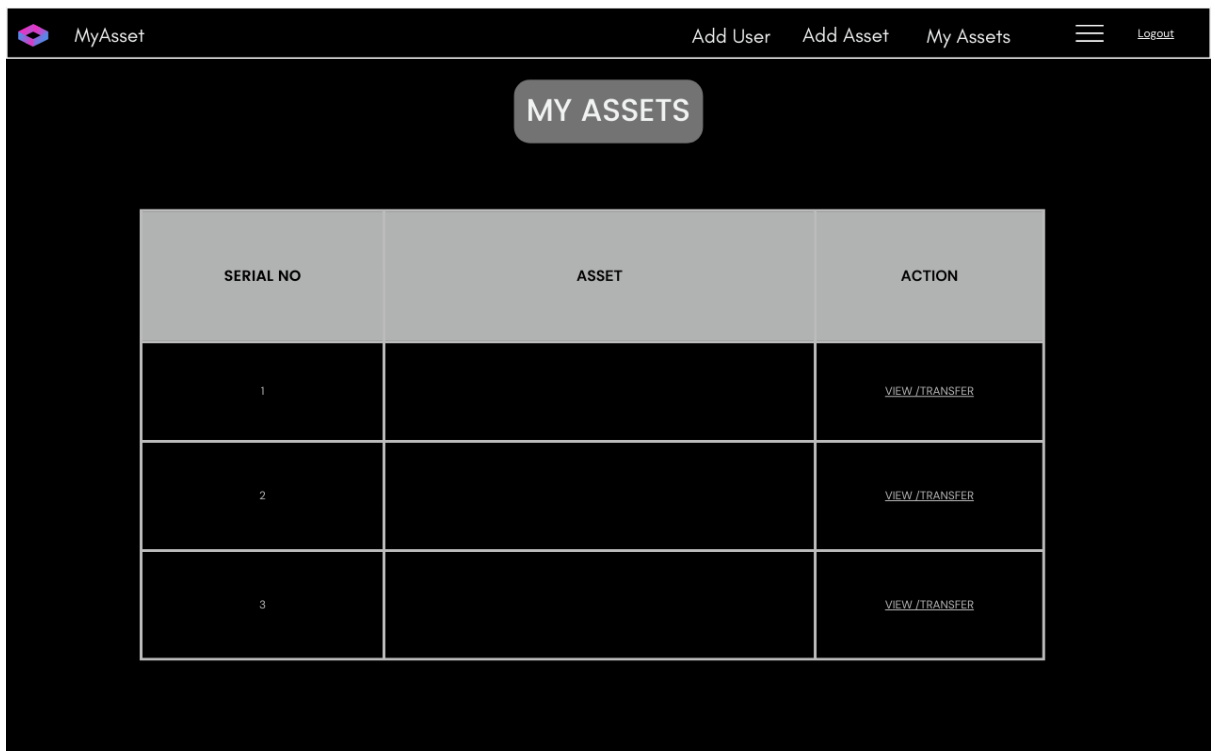
The table will have following fields- Serial No, Udid, IMEI, Action

The action column of each row will contain two buttons- View and Transfer for each asset in the table.

The view button will show the details of respective asset and the transfer button will show the form with some auto filled asset details w.r.t UDID and dropdown list of all the users so that asset can be transferred to that user.

As soon as the admin will transfer the asset, a dialog will pop up with success message and on click it will redirect to my asset page

View, Transfer and Dialog component are part of My Asset Components



SERIAL NO	ASSET	ACTION
1		VIEW / TRANSFER
2		VIEW / TRANSFER
3		VIEW / TRANSFER

MyAsset

Add UserAdd AssetMy AssetsLogout

IPHONE

FMODEL NAME*

Jlara Martins

IMEI*

Jlara Martins

UDID*

hello@infosys.com.com

OWNED BY*

Select

ASSET ID

Jlara Martins

MODEL NO.

Jlara Martins

SERIAL NO.

Jlara Martins

AMANAT ALLOCATION

Jlara Martins

PASSWORD

SIM CARD INSERTED

Select

ADD

Add User

- Add user Page will Display the following input fields-First Name, Last Name, Email-id, Password, a dropdown of an Assign Role with values Admin and users.
- The form will contain mandatory and non-mandatory fields.
- On login the user/admin will be redirected to my asset page.

MyAsset

Add UserAdd AssetMy AssetsLogout

Add User

FIRST NAME*

Jlara Martins

FLAST NAME*

Jlara Martins

EMAIL*

hello@infosys.com.com

PASSWORD

ROLE*

Select

ADD

All Assets

- All Asset tab should have dropdown called Assets-Android, iPad, iPhone,MacMini, MacBook, iMac (display alphabetically). Excel export as per
- dropdown selection from all asset's dropdown value android, iPad, Iphone,MacMini, Macbook, iMac.
- As per dropdown selection, a list of assets should display with mandatoryfields in column (which were entered while adding asset) in column and action buttons.
- Should have action buttons-View, Transfer, Edit, Delete. The action column of each row will contain four buttons -View, Transfer, edit, Deletefor each asset in the table
- As soon as the admin will click on the View, details of particular asset willbe shown and the transfer button will show the form with some asset detailsand dropdown list of all users so that asset can be transferred to that use
- As Soon as the admin will click on the Transfer Button a dialog box will pop up with success message and onclick it will be redirected to my assetspage
- As soon as admin will click on the update button a form with the respectiveasset will be shown for edit options and for the Delete the Complete Row of the Table will be deleted.

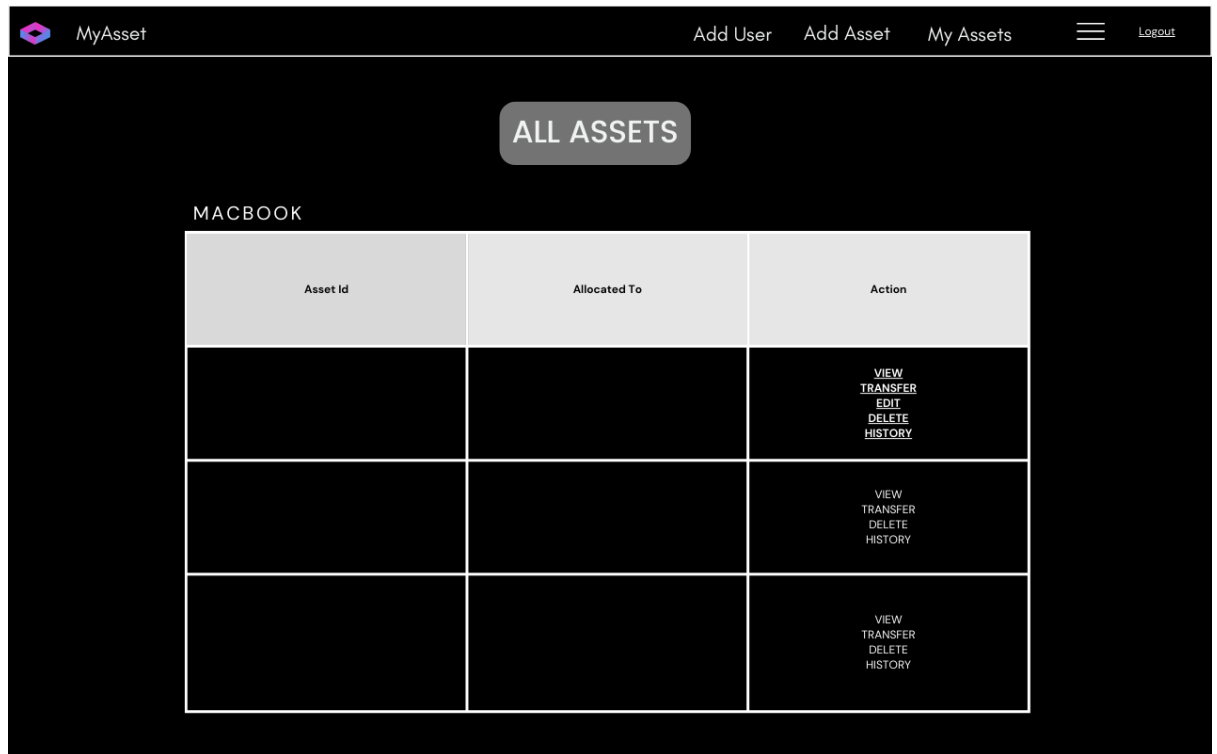
MyAsset Add User Add Asset My Assets Logout

ALL ASSETS

ASSETS*

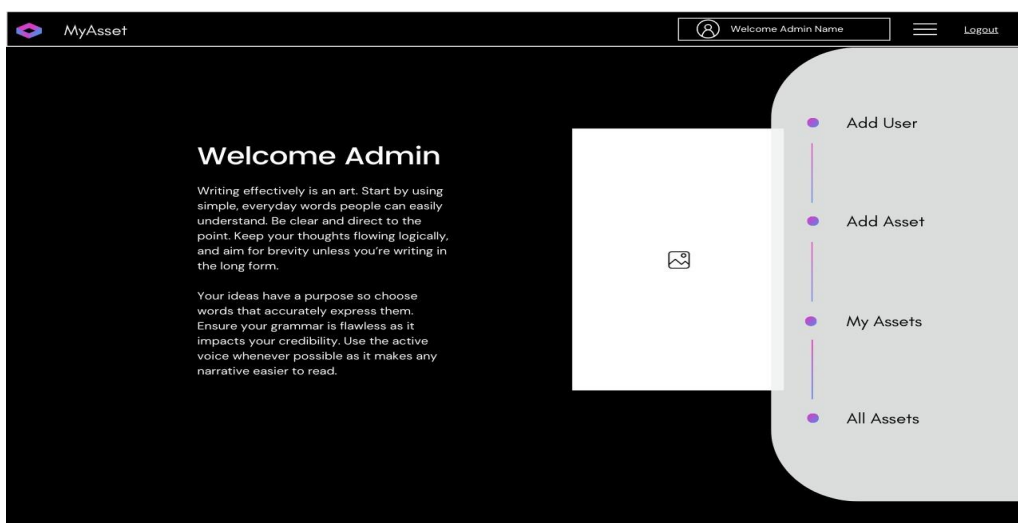
Select

SHOW



Hamburger Menu

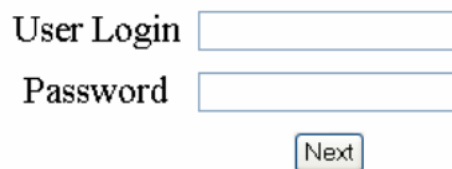
- The hamburger menu will be present on the top left corner of all the screens of web application
- The hamburger menu will contain the following options in the adminportal-Add Asset, Add Users, My assets and all assets
- On clicking on the options admin will be directed to the respective pages and on logout the admin will be logged out and redirected to the Signup/Login page
- Routing for all the components/Pages in Hamburger Menu.



3.5 Computational development

3.5.1 Data Entry from Design

On clicking any of the option buttons in the form, an authentication form as shown in comes up. After filling in User Login and Password and clicking Next button on the authentication form, the corresponding data-entry form for the particular database table comes up. The format of data-entry form for the table User. In the user data-entry form, buttons for add asset will appear button can be used to view the changes after an insert, edit or delete is carried out.

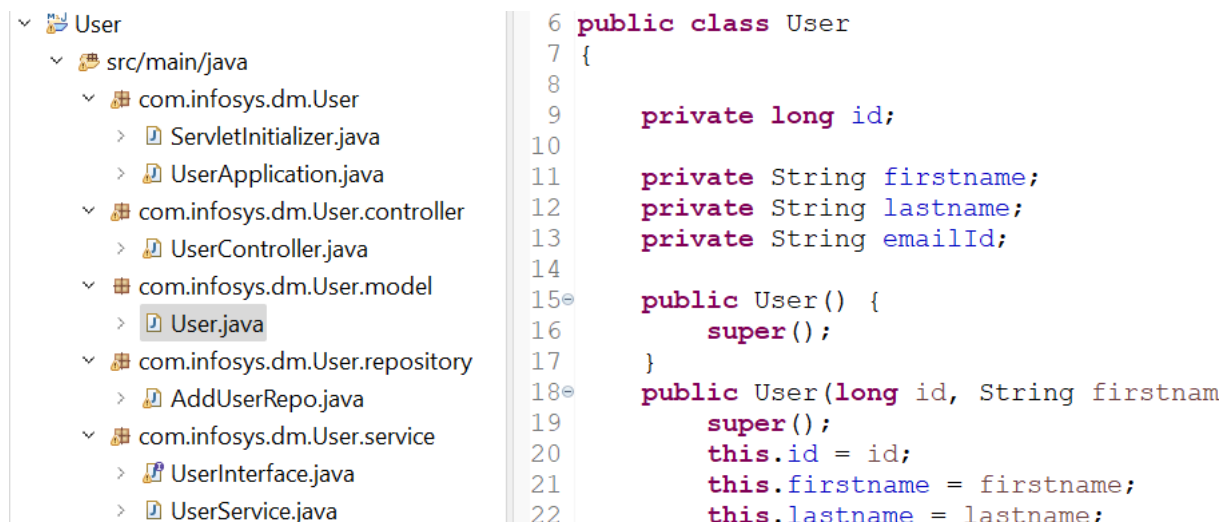


The figure shows a simple login form. It consists of two text input fields. The first field is labeled 'User Login' and the second field is labeled 'Password'. Below these two fields is a button labeled 'Next'.

Fig 3.6 Login Design

3.5.2 Code Breakage:-

Add User it adds the user and help in signing up the user using the Infosys.com domain. The database linkage was done with the backend and security check was done with frontend.



Services:

The Spring boot service component is defined as a class file that includes an `@Service` annotation and allows developers to add business functions. Annotation is also used for classes that provide these business functions. These sections are automatically detected in the spring context where annotation-based setting is used as well as a class-based scan as it provides a special `@Component` annotation. The function performed as part of the

annotations is provided as a visual connector and is independent of the model without the integration mode. This adjective is a general purposeful belief, and engineers may choose to limit the semantics and apply the same as it seems appropriate.

How does it work?

Here we will see the performance of the service component in the Spring Boot, but before that, we need to know not only the value of the service component but also the value of the component of the @Service. It is very important for us to know that standard applications contain different layers, for example, data access, data presentation, services and business, and in each of these layers, there are different beans that allow automatic discovery of layers. @Service annotation helps define classes in the service layer. As mentioned earlier, the @Service component provides the @Component feature. Now, this makes it even more interesting for us to understand that the @Component annotation is used in all applications so that the bean is marked as part of the Spring, and the bean labeled @Service occurs in the business concept component and is therefore only used in the service layer due to additional specialization.

The service component ensures that the class file that includes business ideas is in a separate layer and keeps it separate from the @RestController class file. With the application of the @Service annotation, we need to add spring core dependencies. During the early spring release, all the beans were announced in the XML file, but considering the stiffness indicator, this becomes a herculean function, so the @Component annotation provides an annotation-based injection with a Java-based configuration. Thus, having an injection based on the definition reduces the declaration of a bean as a 'bean marker'. In addition to @Service, which is mostly used in the service layer, @Controller is mostly used in the presentation layer and the @Repository Data Access or DAO or Persistence layer.

We will now look at the performance of a service component using an example. We will announce an interest rate calculator service in the event of a bank deposit. Initially, we will build a service layer, which will contain a service section that will contain the required function. These activities can be simple interests and / or lucrative activities combined. The class will be defined as @Service so that the spring context can detect it automatically, and the pattern can be strengthened from the context. We will now describe a new class that will contain an example of a service class. Application context confirmed. The context.scan () function will now look at all the annotations, and the required annotation will be registered. Now, according to an @Service annotation, we're building a bean for a custom-made service category. This bean will be a model that will be used to summon any activity within the classroom. Once the required tasks are performed, the application context is closed. Bean suspension can be achieved by transparent distribution.

```

1 package com.infosys.dm.User.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5
6
7
8
9
10 @Service
11 public class UserService implements UserInterface
12 {
13
14     @Autowired
15     public AddUserRepo au;
16
17     @Override
18     public void addUser(User us) {
19         au.addUser(us);
20     }
21
22
23     @Override
24     public void getUser() {

```

Controller::

In the Spring Boot, the control section is responsible for processing incoming REST API requests, modifying the model, and retrieving feedback to be provided in response.

The spring categories are explained to @Controller or @RestController annotation. These control sections mark the application holder to allow Spring to be seen as a RESTful service during operation.

In this tutorial, we will include a description of @Controller and annotations of @RestController, their usage conditions, and the differences between the two annotations.

Spring @Controller annotation is also an @ Component annotation. The @Controller annotation indicates that a particular category uses the control role. The Spring Control Annotation is often used in conjunction with the capture methods with annotations based on the @ RequestMapping annotation. It can only be used in classrooms. Used to mark a class as a web application host. It is widely used with Spring MVC applications. This annotation serves as an example of an annotation section, which illustrates its role. The dispatcher scans such classes with annotations in the ways they are mapped and finds annotations @ RequestMapping. Let's understand all of this by example.

The process

- Create a simple Spring Boot project
- Add spring-web dependence to your pom.xml file
- Create one package and name it “controller”
- Create a class within the package
- Use our application within the DemoApplication.java file

```

1 package com.infosys.dm.User.controller;
2
3 import java.util.List;
18
19 @RestController("/user")
20 public class UserController {
21
22     @Autowired
23     private UserInterface userinter;
24
25     // create get all users
26     @GetMapping("/list")
27     public String getAllUsers()
28     {
29         return "sample text";
30         //return userinter.listUser();
31     }
32     // Creating user
33     @PostMapping("/add")
34     public void createUser(@RequestBody User us)
35     {
36         userinter.addUser(us);
37     }
38     // Get user by id
39
40
41 }
42

```

The Main JAVA class that will help to run the program

```

package com.infosys.dm.User;

import org.springframework.beans.factory.annotation.Autowired;

@SpringBootApplication
public class UserApplication extends SpringServletContainerInitializer{

    public static void main(String[] args) {
        SpringApplication.run(UserApplication.class, args);
    }

}

```


CHAPTER 04

PERFORMANCE ANALYSIS

Spring Initializr ::

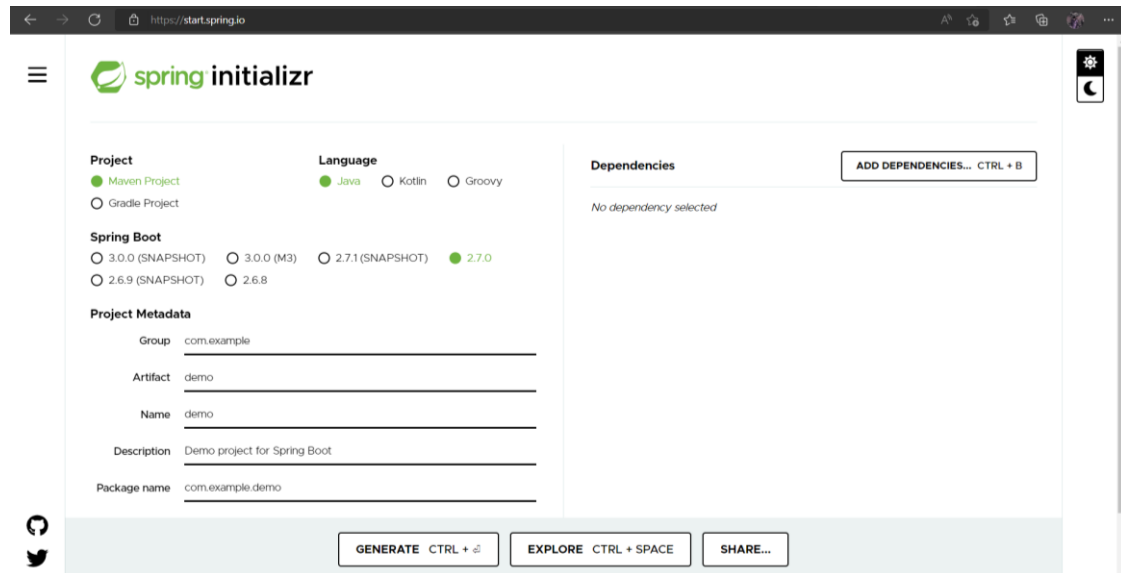
The screenshot shows the Spring Initializr web application in a browser. The URL is https://start.spring.io. The interface is divided into several sections: 'Project' with radio buttons for 'Maven Project' (selected) and 'Gradle Project'; 'Language' with radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'; 'Spring Boot' with radio buttons for versions 3.0.0 (SNAPSHOT), 3.0.0 (M3), 2.7.1 (SNAPSHOT), 2.7.0 (selected), 2.6.9 (SNAPSHOT), and 2.6.8; and 'Project Metadata' with input fields for 'Group' (com.example), 'Artifact' (demo), 'Name' (demo), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo). On the right, there is a 'Dependencies' section with a button 'ADD DEPENDENCIES... CTRL + B' and the text 'No dependency selected'. At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'. The Spring Initializr logo is in the top left, and a settings icon is in the top right.

Fig 4.1 Spring Initializr

The project started out as a small HTML form, which allows you to generate a Spring Boot app with a few options. Welcome to the Spring community, so the team has added new ways to contact you, with your command line or your favorite IDE.

After a while, we decided to work on updating the original UI. We simplified the form, omitted key sections in the front and middle and pushed some into the automatically hidden section. In view of the dependency value, we created a search box to check for available startups.

The project is now a popular service even with a library that you can use and expand according to your needs. These days, the Web UI is not even the most popular client, but we are still committed to working and improving the service.

The Spring Initializr team has been tempted many times to add new features and options, or solve new problems. Do we have to build complex structures to work on project limitations? Should I add more ideas on how to build, distribute or use your Spring Boot app in production? Should we convert it to the original Spring Boot market place?

With this new update, the design often had a (much needed) update and we chose not to show the full list of available dependencies. One can see that list as a function in terms of bad project metadata (we need to improve that!) Or the lack of accessibility on the spring.io website (we need to fix that!).

While recognizing your current limitations, we feel that “promptly updating the newly

installed Spring Boot system” is still the main promise of our service and developers should know what kind of application they want to build when creating a project. However, we think the Spring Boot team can do more to help developers upgrade and update their applications, so we are working to expand what Spring Initializr can do (see previous blog posts).

This new update is a new step towards additional changes on start.spring.io

MySQL::

While the MySQL text model provides the flexibility and precise API that developers love, MySQL information self-control is difficult, time-consuming, and expensive, especially as the application scale. AWS has created Amazon DocumentDB (compatible with MySQL) as a fully-fledged and compliant MongoDB text messaging service that allows you to use your existing MySQL drivers, MySQL clients, and tools with Amazon DocumentDB.

As a fully-fledged AWS data service, Amazon DocumentDB allows you to set up, protect, and measure MySQL-related information in the cloud without having to worry about storing and pasting web software setting up and protecting web collections, using collection management software, configuration, backups, and monitoring of production workloads.

You can transfer MySQL workloads to Amazon DocumentDB using the AWS Database Migration Service (AWS DMS) and command line services like `mongodump` and `mongorestore`.

Benefits of using MySQL Tasks

Amazon DocumentDB supports the MySQL API, and there are some benefits and advantages of using your MySQL load on Amazon DocumentDB.

Scalability

Amazon DocumentDB separates storage by counting, allowing each one to measure independently so you can easily measure reading volume up to millions of requests per second. You can increase your reading capacity to millions of requests per second by adding up to 15 readings per minute, regardless of the size of your data.

High Availability and Stability

Amazon DocumentDB is built for 99.99% availability and duplicates six copies of your data in all three AWS Availability Zones (AZs). In Amazon DocumentDB, continuous backup is automatically enabled, providing 1 day point-in-time recovery (PITR).

Safety and Compliance

Amazon DocumentDB runs on Amazon VPC, allowing you to separate your collection from your virtual network. Amazon DocumentDB supports decryption encryption via AWS KMS, transaction encryption, partial access control, and compliance certificates including PCI DSS, ISO 9001, 27001, 27017, and 27018, SOC 1, 2 and 3, and HITRUST, moreover, HIPAA eligibility.

AWS Service Integration

A DocumentDB easily integrates with other AWS services to expand performance, including A CloudWatch monitoring and alarms, AWS CloudTrail search logs, AWS Glue ETL, and AWS.

Performance Of Spring Boot::

It has sometimes been suggested that the Spring and Spring Boot are "heavy", perhaps because they allow apps to hit more than their own weight, providing many features of the user code that are not too heavy. In this article we focus on memory usage and ask if we can measure the effect of using Spring? In particular we would like to know more about the actual overhead of using Spring compared to other JVM applications. We start by building a basic app with Spring Boot, and look at a few different ways to measure when it works. Then we look at some comparison points: transparent Java applications, applications that use Spring but not Spring Boot, applications that use Spring Boot but do not have automatic configuration, and other sample Ratpack applications.

Vanilla Spring Boot App

As a base we build a consistent application with a few webjars and `spring.resources.enabled = true`. This is great for rendering good looking content that probably has a REST endpoint or two. The source code for the app we used to test is on github. You can build it with mvnw wrapper script if you have JDK 1.8 available also on your way (mvnw package). It can be presented as follows:

```
$ java -Xmx32m -Xss256k -jar target / demo-0.0.1-SNAPSHOT.jarCOPY
```

If we add a load, just to warm up the series pools and force all code methods to be used:

```
$ ab -n 2000 -c 4 http: // localhost: 8080 / COPY
```

We can try and limit the series a bit in application.properties:

```
server.tomcat.max-threads: 4COPY
```

but in the end it does not make much difference to the numbers. We conclude with the analysis below that it will save at least MB in the size of the stack we use. All the Spring Boot webapps we review have this same configuration.

We may need to worry about the size of the class, in order to estimate what is happening in memory. Despite some online claims that JVM memory makes a map of all the pots in the classpath, we actually do not find any evidence that the size of the classpath has any effect on the operating system. For reference, the size of the dependable jars (excluding JDK) for a vanilla sample is 18MB:

```
$ jar -tvf target / demo-0.0.1-SNAPSHOT.jar | grep lib /.*. jar | awk '{tot += $ 1;} QEDA {print tot}'  
18893563 COPY
```

This includes Spring Boot and Actuator launchers, as well as 3 or 4 webjars for static services and a web finder. A small Spring Boot app that combines Spring and some cutting but no web server with about 5MB of pots.

Test Planning:

Test planning is the phase to describe how Testing would be performed. It includes a plan to test before going to start making the test suite. The first step of testing is to test each system module that is once the module has been completed, we test the module. For this, white box testing and black-box testing is used. In white-box testing, structural testing is done so all the modules are tested one by one and finally when the project is completed black box testing is used to test the whole system together. The objective of the system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified and ensure that the computer system and the associated clerical and other procedures work together.

TESTING STRATEGY Following Testing Strategies are used in the Application:

- **UNIT TESTING** Unit testing focuses on the smallest unit of software design, like a module or system component. This testing strategy conducted on each module interface to access Boundary conditions are tested and all error handling paths are tested.
- **INTEGRATION TESTING** This testing strategy follows the testing of combined parts of an application to determine if they function correctly. The purpose of this level of testing is to expose faults in the interaction between integrated units. Testing performed to expose defects in the interfaces and interaction between integrated components

CHAPTER 05

CONCLUSIONS

5.1 Conclusion

5.2 Applications

5.3 Future Scope

5.4 Advantages

5.5 Limitations

5.1 Conclusion

As keeping a track of assets is important for every organization. This application ties the assets with the IT infrastructure of the organization. With a robust asset management system, management and IT professionals can review and monitor all types of assets within the organization. The information can be used to make detailed decisions about purchase and other aspects of the asset's life cycle. The system helps in an accurate record of all types of assets. Using the software can help in efficient resource planning. It can also reduce the risk of theft of assets. An asset management system will help to monitor the assets located in different locations and departments. We will get to know where the assets are located. We can run reports to know about ownership, service details, and other insightful information. The Project provides meaningful information regarding assets to optimize asset utilization and remove manual management tasks of maintaining a spreadsheet for various assets. The featured dashboard can help to get a quick decision by asset count and statistic charts or graphs information.

In this project, we have learnt how to approach an industry level project, used the user stories to design the project and used wide range of complex and flexible features of Angular Framework to implement our Project - "Asset Management Portal" alongside working with spring boot to interact with database using API calls. This project has made us efficient in using latest technologies while implementing the project and handling all the components and modules. We would like to thank Infosys and our mentors for this great learning opportunity.

5.2 Applications

This project can have numerous applications in real world. As can be seen from the example above, instead of storing Students and Subjects in separate databases such as in SQL, here at MySQL, we are storing them together in a single "text". This speeds up the process of data acquisition. Most large apps like Facebook require this small increase in performance because billions of users use the app and therefore, a small increase in performance can have a huge impact on the whole. This is one of the most important benefits of using MySQL.

Other benefits include:

Integration: MySQL allows data sharing across all nodes in the collection to ensure that there is no single point failure on the stored server.

Support for Second Indicators: MySQL allows not only the primary index but also the second important index for most operating systems.

Caching: MySQL saves a lot of data to allow faster detection of query results.

A nice feature set: There are various features in MySQL (Ad-hoc queries, Indicator, Multiplication, Rate Launch, File Storage, Merge, JavaScript extrusion server, snow clusters, etc.) that make it an easy-to-use website.

API testing. Also, it's really comforting for devs to look at their codes, which work in the UI. It was easy to understand things, especially when I returned to the project after a short break. Activating commands from the Swagger UI has always been easier or more understandable.

5.3 FUTURE SCOPE

5.3.1 Integrate A server

It can be used in any govt or private organization so that the website can be fast and storing the data will be easy.

5.3.2 As I am using the Spring Boot it is easy for me or to any other developer or any person who want to use the api to perform CRUD operation easily on the data that is connected to the project.

5.3.3 We can also use these projects in surveys.

5.4 ADVANTAGES

- First, it is based on Java, one of the world's most popular programming languages. Alternatively, Spring Boot can help you quickly build any apps without having to worry about their safe and secure configuration.
- Spring Boot has a large user community which means you can get free reading materials and tutorials. Spring Boot has many strings. This is useful if you are doing long or repetitive tasks. When the main thread is eaten, the others are used simultaneously.
- Other additional benefits include:
- Reduce time spent on development and increase the overall efficiency of the development team.
- Helps automatically adjust all parts of the Spring-grade production grade application.
- It prepares the creation and testing of Java-based applications by providing automated unit setting and integration testing.
- It helps to avoid all manual work boilerplate coding, annotations, and complex XML configuration.
- Comes with HTTP embedded servers like Jetty and Tomcat to test web applications.
- Integrating the Spring Boot with the Spring ecosystem that includes Spring Data, Spring Security, Spring ORM, and Spring JDBC is easy.
- It provides many plugins that developers can use to work with embedded and memory websites smoothly and easily.
- Allows easy connection to online sites and services like Oracle, PostgreSQL, MySQL, MySQL, Redis, Solr, ElasticSearch, Rabbit MQ, ActiveMQ, and many more.
- Provides administrator support - you can manage remotely with the app.

- It makes it easy to learn and comes with the Embedded Servlet Container.
- Provides flexibility in setting up XML setting, Java beans, and Databases.
- Provides easy access to the Command Line Interface which makes the development and testing of Java Boot built with Java or Groovy faster.

5.5 Limitations:

A major challenge that many engineers face when using the Spring Boot is lack of control. Visual style incorporates a number of additional (often unused) dependencies that increase the file size of the feed.

The Spring Boot artifact can be applied directly to Docker containers. This helps to determine when you need to quickly create microservices. However, some engineers argue that since the Spring Boot was designed to be lightweight and fast, it should not be used for monolithic applications.

While Spring Boot comes with basic logging and health monitoring tools for your app, this is not enough. Tools like Retrace help teams easily monitor Java applications. This tool helps detect slow SQL queries, provides performance reports and CPUs and identifies the most common errors in translating logs.

Additionally, it can be very challenging to update your Legacy Spring code. You can overcome this problem by using tools like Spring Boot CLI (Command Line Interface) that will help you change your asset code.

Some of the disadvantages are:

If you have never worked with Spring before and want to learn about proxies, dependence injections, and the AOP program, it is not recommended to start with Spring Boot as it does not include much of this information.

You really need to understand the many basic spring programs (and a little spring history), as well as some advanced topics to fix and solve them.

Spring Boot works well with microservices. Spring Boot materials can also be distributed directly to Docker containers. However, some developers do not recommend a framework for building large and monolithic applications.

If you are unfamiliar with other Spring Ecosystem projects (Spring Security, Spring AMQP, Spring Integration, etc.), using them with Spring Boot will cause you to miss out on many points that you would understand if you started using them independently.

REFERENCE

- Dealt with bugs and errors with the help of YouTube and Stack Overflow.
- Studied about the required libraries/modules from GeeksforGeeks.
- Courses: Infosys Lex course
- Angular: Lex Courses and Angular.io website tutorial
- Java: <https://www.w3schools.com/java>