**Ideapoke Technologies Pvt. Ltd.**





**Technology Stack Document**

Version: P1.0.0

Created by Niharika

Created on: 02-10-2021

**Document Control**

* **Revision history**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** | **Email** |
| 02-10-2021 | V1.0-0 | Initial draft | Niharika | [niharika.ta@ideapoke.com](mailto:sanoj.kumar@ideapoke.com) |
|  |  |  |  |  |
|  |  |  |  |  |

* **Document reviews**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Email** | **Role** | **Date** | **Comments** |
|  |  |  |  |  |
|  |  |  |  |  |

* **Document approval**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Approver** | **Email** |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

1. **INTRODUCTION**…………………………………………………………………………………………....4
   * 1.1 Purpose.…………………………………………………………………………………..….4
   * 1.2 Scope………………………………………………………………………………………....4
2. **Technology Stack Overview** …………………………………………..………………….............5
   * 2.1 Database.………………………………………………………………………………….....5
   * 2.2 Caching………………………………………………………………………………………...7
   * 2.3 Web Server …………………………………………………………………………….......8
     + 2.3.1 Frontend / Graphical User Interface.................................8
     + 2.3.2 Backend.............................................................................9
     + 2.3.3 Tech Stack Diagram..........................................................11
     + 2.3.4 Software Tools for Application.........................................12
     + 2.3.5 Software Tools for Core Engineering................................12
     + 2.3.6 Software Libraries for Application....................................12
     + 2.3.7 Software Libraries for Core Engineering...........................13
     + 2.3.8 Authentication....................................................................13
     + 2.3.9 Map.....................................................................................14
     + 2.3.10 Hosting Server...................................................................14
   * 2.4 Version Control ……………………………………………………………………....…....15
   * 2.5 Deployment....……………………………………………………………………....…......16
3. **Design and Architecture**.......................................................................................17
4. **API Documentation**...............................................................................................19
5. **Security**.................................................................................................................20
6. **Installation**............................................................................................................21
7. **INTRODUCTION**

A technology stack, also called a solutions stack, is a list of all the technology services used to build and run one single application.

**1.1 Purpose of the Document:**

A Tech Stack is a set of tools that are used to construct and power an application. It consists of a combination of software applications, frameworks, and programming languages that realize some aspects of the program.

It’s important for a company to choose a set of technologies that are capable of developing both the user-side (front-end) and server-side (back-end) of their application.

**1.2 Scope of the Document:**

Scope of this document is to make a list all technologies we have used and write a short description for each and why we have chosen it.

Structure-wise, tech stack consists of two equal elements. One is frontend or client-side; the other is server-side or backend. Combined, they create a stack

1. **TECHNOLOGY STACK OVERVIEW**

Each app’s infrastructure should include:

* **A database**—to store app data.
* **A caching system**—to decrease database load, helping the database process surges in traffic.
* **A Web server**—to handle requests sent from the browser and return information.

**2.1. Database**

* The function of a database is to search, filter, present, and sort the information in a Web application based on user requests sent via a Web browser.
* To further support user queries, a database can perform various calculations. Databases are the backbone of many online services, e.g., e-commerce websites or discussion boards.
* We are using relational & non-relational database in our application.

**MySQL (https://www.mysql.com/) :**

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications

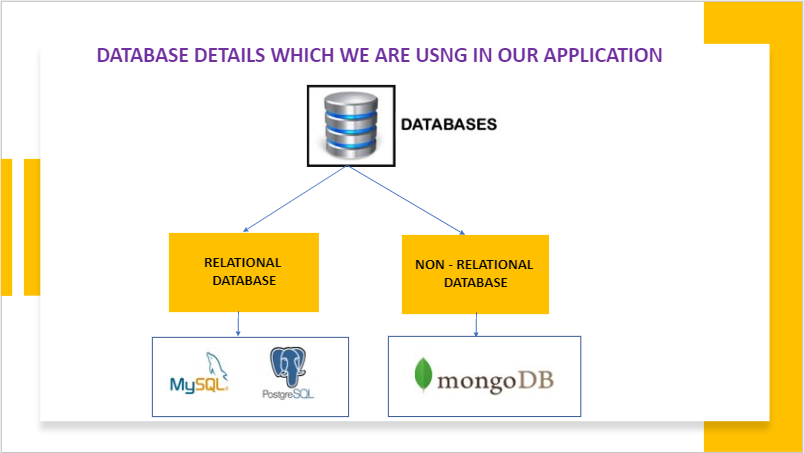
**PostgreSQL (**[**http://www.postgresql.org/**](http://www.postgresql.org/)**) :**

PostgreSQL is a powerful, open source object-relational database system. We use it in deployed version at Heroku, because this the main choice and it works good. We didn’t want to complicate with NoSQL databases, because our data scheme is very simple and SQL works good enough.

**MongoDB (**[**https://www.mongodb.com/**](https://www.mongodb.com/)**):**

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema

The below diagram shows the database details using in our application.



**2.2 Caching System**

* Caching system reduces server lag and relieves the database of heavy loads during peaks in traffic by serving previously used and accessed data.
* Without the caching system, Web apps are often sluggish and offer poor site performance.
* We are using **Redis caching mechanism** in our application to cache the data.
* **Redis** is an open source (BSD licensed), in-memory data structure store, used as a database, **cache** and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyper loglogs, geospatial indexes with radius queries and streams.



**2.3 Web Server**

A Web server receives, processes, and delivers user (i.e. a Web browser) requests in the form of an HTML message.

There are two sides of web development:

* + The client side / Front end
  + The server side / Backend

## **2.3.1 Frontend / Graphical User Interface:**

The frontend / the client-side, encompasses everything a user can see. The frontend makes the interaction with a Web application possible through the tangible presentation layer.

Here are the major frontend technology stack components:

* + **HTML 5.0 (Hypertext Markup Language)** — responsible for the structure of the displayed content.
  + **CSS 3.0 (Cascading Style Sheets)** — describes the style (presentation) of the content (layout, colors, fonts, spacing), can be short-handed with tools such as SASS.
  + **JavaScript** —breathes interactivity into a Web application: animations, forms, image carousels, visuals, etc.
  + **jQuery 3.3** ([**http://jquery.com/**](http://jquery.com/)) — jQuery is a fast, small, and feature-rich JavaScript library. We had to use it because jQuery is dependency for Twitter Bootstrap. We have also used it for making AJAX requests.
  + **AJAX (Asynchronous JavaScript and XML)** — Used for updating only snippets in a web application without reloading the whole page.
  + **Bootstrap 3.0 (http://getbootstrap.com/)** — Responsive design makes it possible for a web page.
  + **React JS:** React is a JavaScript library created for building fast and interactive user interfaces for web and mobile applications. It is an open-source, component-based, front-end library responsible only for the application’s view layer.

**Design Tools:**

Photoshop and Illustrator.

**2.3.2 Backend**

The backend / the server-side, governs the logic behind how a website works. Essentially, the backend is the technology responsible for the communication between the client-side, the server-side, and all the components within the backend.

Here are the major backend technology stack components:

**PROGRAMMING LANGUAGE :**

* + **Java 1.8 (**[**https://www.java.com/en/**](https://www.java.com/en/)**)**— Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible
  + **Phyton 3.5 (**[**https://www.python.org/**](https://www.python.org/)**)**— Python is an interpreted, high-level, general-purpose programming language. **Python** can be **used** to develop different applications like web applications, graphic user interface based applications, software development application, scientific and numeric applications, network programming, Games and 3D applications and other business applications.

**Frameworks:**

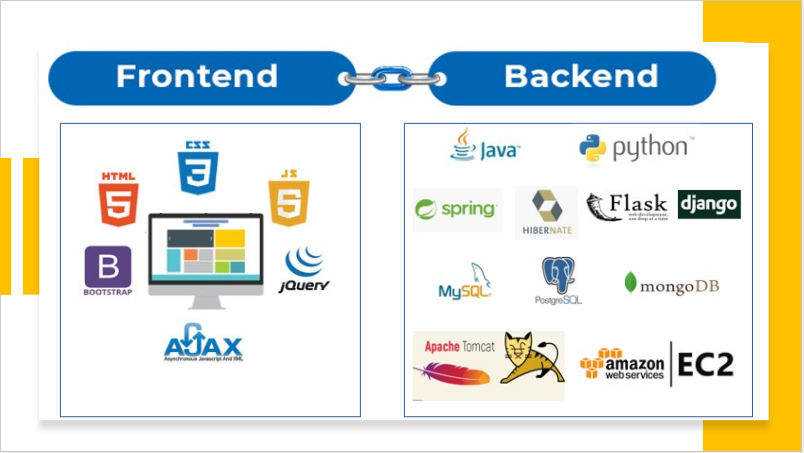
* + **Spring 4.0 (**[**https://spring.io/**](https://spring.io/)**)** — Spring is an enterprise Java framework. It was designed to simplify Java EE development and make developers more productive. Spring makes use of Inversion of Control and Dependency Injection to promote good software coding practices and speed up development time.
  + **Hibernate 3.0 (**[**https://hibernate.org/**](https://hibernate.org/)**)** — Hibernate ORM is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database.
  + **Django (**[**https://www.djangoproject.com/**](https://www.djangoproject.com/)**)** — Django REST framework is a powerful and flexible toolkit that makes it easy to build Web APIs. It has a really good web browsable API which can help you, when you develop API. Documentation is also very good and community is active.
  + **Flask (**[**https://flask.palletsprojects.com/**](https://flask.palletsprojects.com/)**)** — Flask is a micro web framework written in Python. It is classified as a microframework because it does not require tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions
  + **Spring Boot (**[**https://spring.io/projects/spring-boot**](https://spring.io/projects/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.

**OTHER TECHNOLOGIES :**

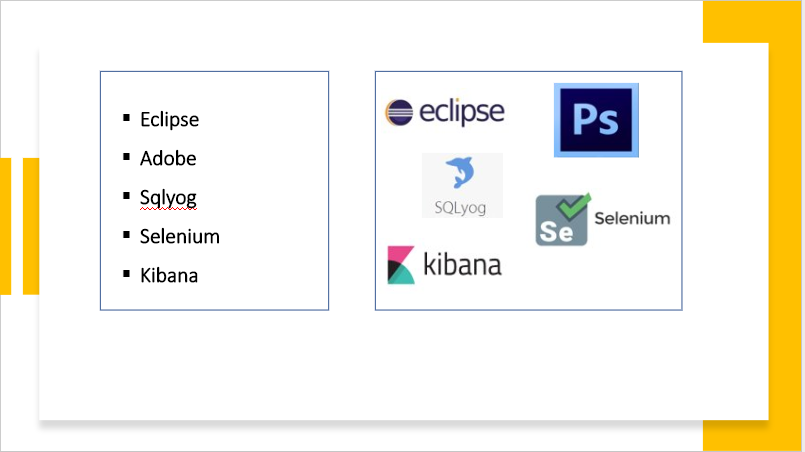
* + **Tomcat 8.5 (http://tomcat.apache.org/)**— The Apache Tomcat® software is an open-source implementation of the Java Servlet, Java Server Pages, Java Expression Language and Java WebSocket technologies.
  + **Elastic 7.4.2 (**[**https://www.elastic.co/**](https://www.elastic.co/)**)** — Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents

**2.3.3 TECH STACK DIAGRAM :**

The below diagram shows frontend and backend technologies which we are using in our application.



**2.3.4 Software Tools For Application**



**2.3.5 Software Tools For Core Engineering**



**2.3.6 Software Libraries For Application**

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Purpose** |
| Slf4j | 1.2 |  |
| Apache POI | 3.16 | To read excel file |
| Java Mail |  | To send mail |
| Jsoup | 1.7.3 | For scrapping |
| Convert api | 2.2 | To convert PPT to PDF, various file formats conversion also possible from this library |
| Itextpdf | 5.5.8 |  |
| Joda-time | 2.9.5 | Date and time library for Java |

**2.3.7 Authentication**

Authentication is the process of verifying the identity of a person or device

* + **Email Authentication -** Email authentication is a technical solution to proving that an email is not forged. In other words, it provides a way to verify that an email comes from who it claims to be from
  + **Password Authentication -** password-based authentication used to validate users. We are using **MD5 algorithm** is used as an encryption or fingerprint function for a file. Often used to encrypt database passwords, MD5 is also able to generate a file thumbprint to ensure that a file is identical after a transfer for example

**2.3.8 Map**

* + **Geo-IP -** Google Analytics is one of the most popular digital analytics software. It is Google's free web analytics service that allows you to analyze in-depth detail about the visitors on your website.

**2.3.9 Hosting Server**

A web application needs a **server** to handle requests from clients’ computers.

We are using **Amazon EC2 – Server** for our web application.



**2.4 Version Control**

* + Repository Code version control tool.
  + Bug tracking and project management tool

**GitHub (**[**https://github.com/**](https://github.com/)**)** —

* + GitHub is a web-based hosting service for software development projects that use the Git revision control system. We decided for it mainly because of git. It is really good tool for source control and also team work.
  + Team members can work on files and easily merge their changes in with the master branch of the project



**JIRA (**[**https://id.atlassian.com/login**](https://id.atlassian.com/login)[**/**](https://github.com/)**)** —

* + Jira is a is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management.
  + It can be used for various use cases - from managing project tasks, to tracking bugs, to doing customer support



**2.5 Deployment**

Software deployment is all of the activities that make a software system available for use. The general deployment process consists of several interrelated activities with possible transitions between them. These activities can occur at the producer side or at the consumer side or both

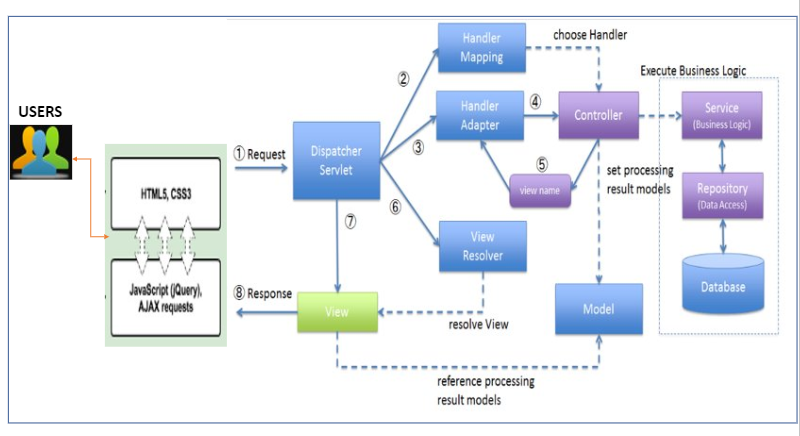
* + Building the class files by using **Maven build** commands
  + To review the code using **sonar lint** tool.

Pipeline Technologies :

* + Auto build by using **Jenkins**

1. **DESIGN AND ARCHITECTURE**

Our application is a typical client-server application. We have provided a scheme bellow, where you can see the most important parts of the application.



## **Server side**

Java & Spring MVC Framework

This server contains **Spring MVC & Hibernate framework**. Spring MVC has a model-view-controller (MVC) architecture:

* **Model** - A model contains the data of the application. A data can be a single object or a collection of objects.
* **Controller** - A controller contains the business logic of an application. Here, the @Controller annotation is used to mark the class as the controller.
* **View** - A view represents the provided information in a particular format. Generally, JSP+JSTL is used to create a view page. Although spring also supports other view technologies such as Apache Velocity, Thymeleaf and FreeMarker.
* **Front Controller** - In Spring Web MVC, the DispatcherServlet class works as the front controller. It is responsible to manage the flow of the Spring MVC application.

**Spring 3 layers**

* Controller – Take the request & gives the response
* Service – Business logic
* Interface
* Impl class
* Dao – Data base connection
* Interface
* Impl class

**Client Side**

HTML, CSS, Bootstrap, JavaScript & jQuery, React JS

**Database**

MySQL & MongoDB

1. **API DOCUMENTATION**

* Rest Template is used to create applications that consume RESTful Web Services.
* Spring Boot is an open-source Java-based framework used to create a micro-Service.​

|  |  |
| --- | --- |
| **PROJECT NAME** | API URL |
| EnterpriseCommonService | http://localhost:8090 |
| EnterpriseDataSetService | http://localhost:8091 |
| PostRequirementService | http://localhost:8023 |
| EIMSService | http://localhost:8094 |
|  |  |

1. **SECURITY**

Authorization is a security mechanism used to determine user/client privileges or access levels related to system resources, including computer programs, files, services, data and application features

* Server level security
* Database level security
* Browser level security

1. **INSTALLATION**

## **How to setup a development environment**

Here are the instructions, how to setup a development environment.

The following **prerequisites** are required to run application

* Java 1.8
* Tomcat 8.5
* MySQL 5.1
* Git
* Maven
* Eclipse
* MySQLYog

Reference link: <https://ideapoketech.sharepoint.com/:w:/s/DevelopmentTeam/EZCaOCMMLlRKmPzDCu9XLLwB_Uz5UHVHrZ-TgQa0Kt4jow?e=FOaBh8>