

**Website Builder
Software Intern, Paperplane**

DATE: 7TH FEBRUARY 2022- PRESENT

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

Bachelor of Technology

In

Computer Science and Engineering

By: Shashwat Kapoor

181418

Under the supervision of

Mr. Varun Goyal, Mr. Devansh Swarup

Dr. Vipul Sharma



**Department of Computer Science & Engineering And Information
Technology**

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT,
SOLAN,**

HIMACHAL PRADESH – 173234

CERTIFICATE

This is to certify that the work which is being presented in the internship report titled “Website Builder Software Intern, Paperplane ” in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Wagnaghat, is an authentic record of work carried out by **Shashwat Kapoor** during the said period; February 2022 – till date, ensuring proper care towards the rules and regulations as specified by the Non-Disclosure Agreement signed between Shashwat Kapoor and Paperlane Pvt Ltd.,Bangalore, , dated 10 February, 2022.

Shashwat Kapoor

181418

Jaypee University of Information Technology Wagnaghat, Solan, H.P. The above statement made is correct to the best of our knowledge.

Mr. Varun Goyal(CTO)

Mr. Devansh Swarup(CEO)

Paperplane Pvt Ltd. Bangalore, Punjab.

Dr.Vipul Sharma (Assistant Professor)

Jaypee University of Information Technology Wagnaghat, Solan, H.P

ACKNOWLEDGEMENT

This is a matter of pleasure for me to acknowledge my deep sense of gratitude to my college, Jaypee University of Information Technology for giving me an opportunity to explore my abilities via this internship program. I would like to express my sincere gratitude to our Training and Placement officer, Mr. Pankaj Kumar and our faculty Coordinator, Dr. Nafis U Khan for this opportunity. I also wish to express my gratitude to my internship supervisors, for their valuable guidance and advice towards my internship.

I would like to record my sincere appreciation and gratitude towards all the officials, coaches, trainers, mentors and employees of Paperplane Pvt Ltd. , without whose kind assistance, my internship program would not have been proceeding in a swift direction. The facts and other vital information provided by them have contributed towards making this report as comprehensive as possible. I am indeed thankful to them.

Last but not the least, I would like to express my sincere thanks to all my family members, friends and well-wishers for their immense support and best wishes throughout the internship *duration* and the preparation of this report and I wish they would continue to contribute towards my well-being.

I believe that this report will be a valuable asset not only for academic institutions, but will also be useful for all those who are interested to learn about internship experiences in healthtech firms.

Shashwat Kapoor

181418

Jaypee University of Information Technology,

Waknaghat, Solan, H.P

CANDIDATE'S DECLARATION

I, the undersigned solemnly declare that the internship report is based on my own work carried out during the course of my work as Software Engineer - Intern at Paperplane Pvt. Ltd.

I assert the statements made and conclusions drawn are an outcome of my own analysis and work.

I further certify that:

1. The work contained in this report is original and has been done by me.
2. The work has not been submitted by any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
3. I have followed the guidelines provided by the university in writing the report. 4.

Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credits to them in the text of the report and given their details in the references.

Shaswhat Kapoor

181418

Jaypee University of Information Technology

Waknaghat, Solan, H.P

INTERNSHIP REPORT UNDERTAKING

I, Shashwat Kapoor, Roll No. 181418, Branch – Computer Science and Engineering currently pursuing my internship with Paperpane Pvt. Ltd from 07th February, 2022 to July, 2022.

As per procedure I have to submit my internship report to the university related to my work that I have done during this internship.

I have compiled my internship report, my mentor in the company is not able to sign this report and no digital signatures are allowed as part of the company's confidentiality policy.

So, I hereby declare that the internship report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to the Training and Placement Cell whenever COVID-19 situation gets normal.

Shashwat Kapoor

181276

Jaypee University of Information Technology

Waknaghat, Solan, H.P

TABLE OF CONTENT

Content	Page No.
CERTIFICATE	I
ACKNOWLEDGEMENT	II
ABSTRACT	III
1. Introduction	1-4 1.1
Paperplane Introduction	1-2
2. Description of the Industry.....	5- 9
1.1 HealthTech.....	5-6 1.3
1.2 Website Builder.....	6
3. Tools and Technologies.....	10-27
2.1 Overview	10
2.2 Java and Frameworks	10 - 21
2.3 MRF Stack	21 - 27
4. Project Implementation	28-33
5. Conclusion	34
6. Reference	35

Chapter 01 : INTRODUCTION

1.1 Introduction

Paperplane

Paperplane is a Bengaluru based health-tech startup that helps doctors digitize their clinic and help them build their own digital clinic brand. The company was started in 2021 amidst the Covid-19 pandemic to help provide quality health care to everyone. Since then, Paperplane has raised 400,000\$ from VC firms across the country. Paperplane will build a digital clinic and virtual receptionist on WhatsApp to enable physicians around the world to leverage their personal brands to provide patients with the most personalized medical experience.

The internship was with Paperplane . The current product offerings include an app, pharmacy application along with web dashboard and portfolio websites. The given project was to make a templated website builder. This was accompanied with major development and bug fixes on the client facing frontend dashboard. The frontend dashboard is built on ReactJs while the backend is made on FastApi. While most of the work involved working on the frontend, this internship also provided a good learning opportunity to learn database designing for large scale projects. The database used was MongoDB while the mobile development was done with Flutter.

The major focus was on successfully understanding their working structure and pattern alongside gaining an extensive skill set as Full Stack Engineer and related underlying technologies, the focus was on learning soft skills like communicating within a corporate firm and working with the team.

Internship goals included successfully understanding various coding paradigms used by a company to build its application and get thorough understanding of some of the company's existing products and some of the upcoming products and projects.

The working culture of the company is great. Paperplane has a typical blend of work and fun. Internship included a fair bit of communication through virtual technologies like meet, slack.

The project undertaken during this period presented a great learning opportunity and practice to work under pressure as well as urgency to work in real time live client projects. In Any case, in the task that needed to put on a certain something, what this opportunity provided me with most is the way that this chance truly opened the entry way into my profession. The amazing thing about this temporary position experience is that it truly overcame any barrier between learning things and really applying a portion of this information into a reality project and getting appreciated for it. The way that this entry-level position straightforwardly identified with what was required to get started on company projects from contributing to actually owning them was of immense help. The internship at Paperplane, Bangalore was a great opportunity to learn, test and work on Full Stack Engineering skills alongside working on soft skills like communication.

Chapter 02 : DESCRIPTION OF THE INDUSTRY

2.1 HealthTech (Health & technology)

The term HealthTech is made up of two words Health and Technology, it is basically the aggregation of the two fields where the technology is used to innovate and improve newer domains into the world of health care. From the ability for consumers or users to view and take their practice online, to applications that allows them to make and manage everyday operations of their clinic. HealthTech's closest examples in our day to day life would be implementation of apps like Practo. On one hand, where apps like Practo help doctors with discovery, Paperplane helps doctors in their digital marketing, build an online presence, and expand their clientele to pan India.

One of the biggest opportunities provided by the advent of healthtech in the current marketplace is in terms of aiding the common man with the opportunity of having quality healthcare from anywhere in the world. This is specifically important for Tier 2 and Tier 3 cities where quality and quick healthcare was beyond the reach of the common man.

HealthTech's growth can primarily be attributed to the Covid-19 pandemic, where the healthcare systems of major countries collapsed to the brink. Moreover, the solutions offered by fintech companies are no longer "universal." Instead, it provides targeted (often niche) services that fill gaps in specific health care needs. It may also be offered at a much lower cost than the services offered by traditional health care providers.

2.2 Paperplane

Paperplane offers multiple products and services in domains of the payment be it in terms of digital marketing, invoice generation or simple appointment booking and management.

The current team under which this particular internship is working on the dashboard product and website builder of the company. The following projects include a basic version for small clinics and individual practitioners and an enterprise version for hospitals and polyclinics along with some additional products like digital pharmacy and website builder

Following are the major projects which are currently operating :-

1. **Web Dashboard** : The dashboard is built on ReactJs. It is the core product offering of the company where doctors and their staff can book and manage appointments. The doctors can also create digital prescription. It also provides them with personally curated marketing content & posters which they can share to build and expand their digital presence. The clients can also view and track their online transactions along with the ability to make a custom google-form like web based app for patient feedback.

Dashboard link: <https://app.paperplane.health>

2. **Mobile Application**: The mobile app, available on Android & Ios is built on the flutter platform. The app offers the same features as the web dashboard along with some additions like the option to send digital prescriptions and chat with patients on Whatsapp.

3. **Pharmacy:** This feature is basically an integrated solution with core product offerings of the company to provide enterprise clients with a holistic experience. It's a web-based light application to handle lab tests and pharmacy orders which currently can handle a small database of 3000-5000 medicines, lab tests and other offerings.

4. **Website Builder:** This product helps individual practitioners and small clinics by helping them build their portfolio clinic websites. The frontend of this project was made in React Js while the automated flows were built on FastApi, github workflows and terraform helped in hosting these sites within 5 minutes.

Chapter 03 : Tools and Technologies

3.1 Overview

The internship at Paxcom (Paymentus) involved working in capacity of Full Stack Engineer learning skills in languages and tools like github workflows and Terraform alongside working on Web Technologies like MRF (MongoDB, Express, React, NodeJS) Stack with HTML, CSS, github.

Apart from all this, this internship also had a focus on soft skills and team work management tools, the agile methodology to work and with daily scrum to align the project requirements and functionalities to tasks across teams. The constant iterations of the code reviews and code fixes were essential in building a better knowledge base of the said technology alongside of getting a grasp of good coding practices in line with the set practices of the code already established in the company's codebase.

The following technologies were used in this project :

3.2 Java

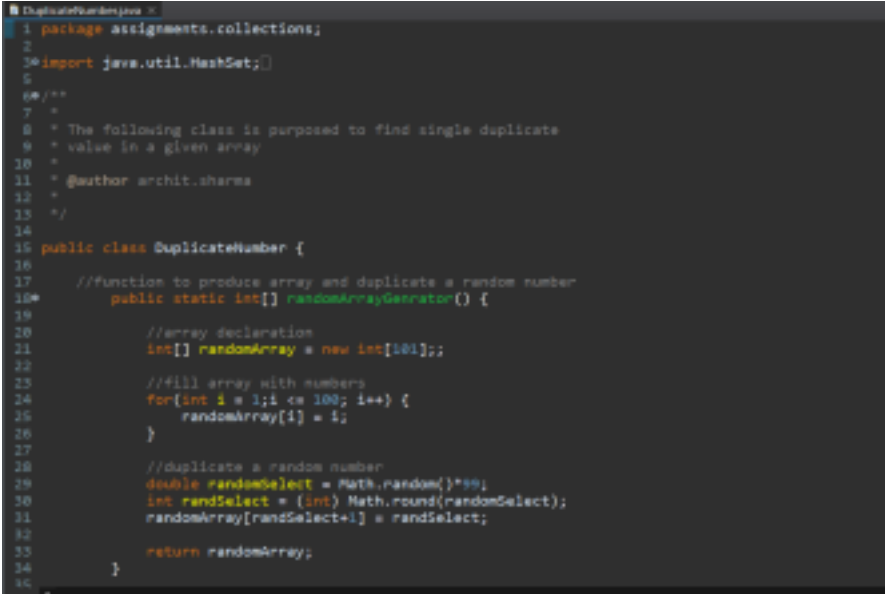
Java is a powerful and vast general-purpose programming language Java which is primarily used in desktop and mobile application development, big data processing, embedded systems, and many other fields. Java is one of languages that support OOPS(Object Oriented Programming Systems) and describes everything in terms of classes and objects hence it is highly viable in building real world systems. Java has a wide variety of frameworks to choose from while building powerful enterprise level code and scalable systems.

Major Applications include :

- Mobile Application Development
- Desktop Application Development
- Web Apps
- Web Servers
- Games etc.

Java gains its popularity and wide applications due to following major reasons why java is used :

- It is based majorly on OOPS(Object Oriented Programming Systems) paradigm which makes it easy to use and conceptualize.
- It has a great community support
- Supports concepts like garabage collection etc.



```

1 package assignments.collections;
2
3 import java.util.HashSet;
4
5
6 /**
7  *
8  * The following class is purposed to find single duplicate
9  * value in a given array
10 *
11 * @author archit.sharma
12 *
13 */
14
15 public class DuplicateNumber {
16
17     //function to produce array and duplicates a random number
18     public static int[] randomArrayGenerator() {
19
20         //array declaration
21         int[] randomArray = new int[101];
22
23         //fill array with numbers
24         for(int i = 1; i <= 100; i++) {
25             randomArray[i] = i;
26         }
27
28         //duplicate a random number
29         double randomSelect = Math.random()*99;
30         int randSelect = (int) Math.round(randomSelect);
31         randomArray[randSelect+1] = randSelect;
32
33         return randomArray;
34     }
35 }

```

Fig 1 : Typical java code

3.2.1 JDBC (Java database connectivity)

JDBC (Java database connectivity) is an API (application programming interface) used in Java programming to interact with a database. JDBC classes and interfaces allow an application to send a request from a user to a specified database.

```

1 package jdbcpractice;
2
3 import java.sql.Connection;
4
5
6
7
8 public class JDBCtestpractice {
9     static final String DB_URL = "jdbc:mysql://localhost/";
10    static final String USER = "root";
11    static final String PASS = "archit04";
12
13    public static void main(String[] args) {
14        // Open a connection
15        try(Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
16            Statement stmt = conn.createStatement();
17        ) {
18            String sql = "CREATE DATABASE STUDENTS";
19            stmt.executeUpdate(sql);
20            System.out.println("Database created successfully...");
21        } catch (SQLException e) {
22            e.printStackTrace();
23        }
24    }
25 }

```

Fig 2 : Typical JDBC query to create a database using JDBC

3.2.2 Log4j

Logging is an essential feature of any given project, The purpose of the log is to act as a warning sign when something bad happens. Regularly reviewing the logs will help detect malicious attacks on your system. Given the large amount of log data generated by the system, it is not practical to manually inspect all these logs daily. Java supports logging using log4j2. The logging can be done on multiple platforms or the configurations to append logs on both local file as well as system console, in log4j we can configure what severity of issue we want to log be it error, warning and other issues etc.

```

# Define the root logger with appender X
log4j.rootLogger = DEBUG, X

# Set the appender named X to be a File appender
log4j.appender.X=org.apache.log4j.FileAppender

# Define the layout for X appender
log4j.appender.X.layout=org.apache.log4j.PatternLayout
log4j.appender.X.layout.conversionPattern=%m%n

```

Fig 3 : Log4j Configuration

```

import org.apache.log4j.Logger;

import java.io.*;
import java.sql.SQLException;
import java.util.*;

public class Example{

    /* Get the class name to be printed on */
    static Logger log = Logger.getLogger(Example.class.getName());

    public static void main(String[] args)throws IOException,SQLException{
        log.debug("Hello this is a debug message");
        log.info("Hello this is an info message");
    }
}

```

Fig 4 : Log4j Example

3.2.3 Hibernate

Hibernate is a Java framework that eases the development of Java apps for interacting with databases. This is a lightweight open source ORM (Object-Relational Mapping) tool. Hibernate implements the JPA (Java Persistence API) specification for data persistence. Hibernate is a high-performance object / relational persistence and query service.

In addition to mapping Java classes to database tables (and Java data types to SQL data types), Hibernate also allows the system to query and retrieve data in the form of models without explicitly typing the underline database's query language.

It basically contains an abstraction layer and handles the implementations contained within it. The implementation includes querying for CRUD (Create Read, Update, Delete) activity, connecting to the database, and other tasks. Hibernate creates persistent logic to store and process data for the long term use.

The major advantage of hibernate system consists when we dont want to change the entire system laguage while working across database and also in use cases where the database needs to be changed, modified oe even updated,

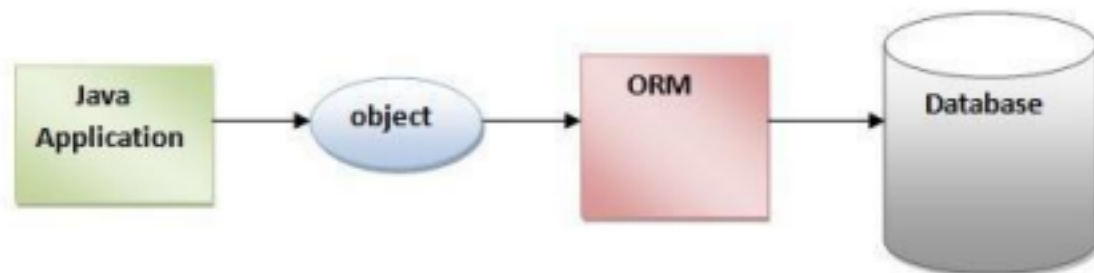


Fig 5 : Interaction with database using Hlberate layer

Hibernate converts a given class into a database model / table using specific annotations which maps the given class and its member data to a said specified role.

```

@Entity
@Table(name = "EMPLOYEE")
public class Employee {
    @Id @GeneratedValue
    @Column(name = "id")
    private int id;
  
```

Fig 6: Hibernate annotations


```

package com.control;

import org.hibernate.Query;

public class HibernateMain {

    public static void main(String[] args) {

        try {

            Configuration configuration = new Configuration().configure();
            configuration.addAnnotatedClass(com.hibernate.Player.class);
            StandardServiceRegistryBuilder builder = new StandardServiceRegistryBuilder()
                .applySettings(configuration.getProperties());

            SessionFactory factory = configuration.buildSessionFactory(builder.build());

            Session session = factory.openSession();

            Transaction transaction = session.beginTransaction();

            Query<Player> query = session.createQuery("from Player where playerName=:name");
            query.setString("name", "po");
            List<Player> playerList = query.list();
            for (Player p : playerList) {
                System.out.println(p);
            }

            transaction.commit();
            session.close();

        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }

    }

}

```

Fig 7 : Typical hibernate code to fetch records from table based on name

3.2.4 Spring Framework

Spring is basically a framework in java which is very powerful in its functionality and applications, mostly used to build enterprise level applications in java

It more commonly referred to as a framework of frameworks of framework because of its wide variety of range of underlying frameworks/ functionalities in multiple domains. These functionalities can broadly be divided into following 6 modules:

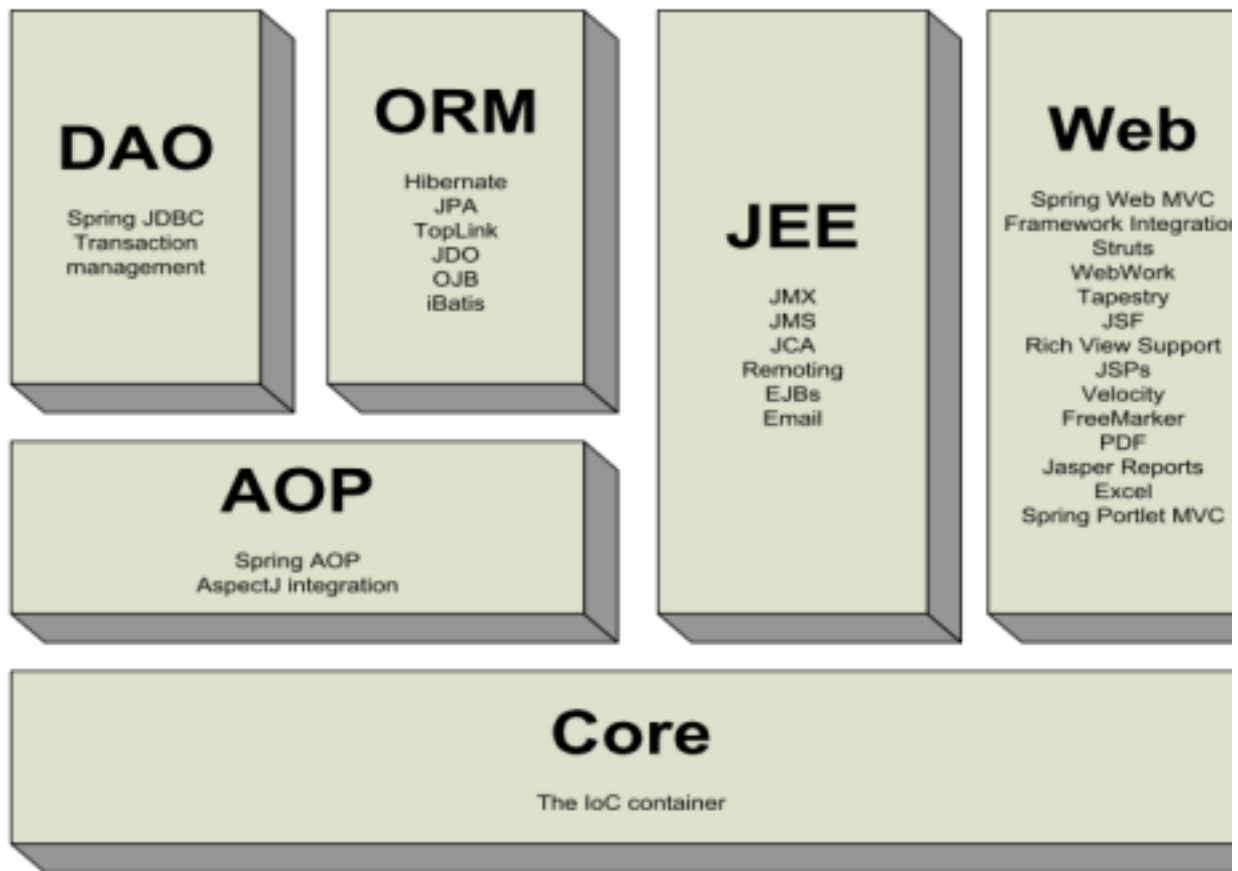


Fig 8 : Spring Framework and its modules.

3.2.4.1 Core

The core module is the most important as well as fundamental part of the framework, it contains the central concepts like the IoC and Dependency Injection. The most basic and essential concept here was of BeanFactory, which in turn provides a sophisticated implementation of the pattern factory which basically changes the requirement of the programmatic singletons where it needs to be added hence there forth decoupling the configuration and specification of all the dependencies from the actual business logic decoupled from the programmatic end.

3.2.4.2 Context

The Context package is build on the base that is the Core packages and works

on strong foundation provided by the core. It basically provides a method to access objects in a very framework oriented manner. The context package inherit its features from the packages like bean and then latter adds other support to it

3.2.4.2 DAO

The DAO (Data Access Object) provides a JDBC (Java Database Connectivity) abstraction layer that basically removes the need to manually work on accessing the data through JDBC(Java Database Connectivity) connection.

The DAO (Data Access Object) package provides a JDBC(Java Database Connectivity) abstraction layer that removes the need to do tedious JDBC (Java Database Connectivity) coding and JDBC (Java Database Connectivity). package provides a way to do declarative transaction management in a a programmatic as well as declarativespecial interfaces, POJO (Plain Old Java Object).

3.2.4.2 ORM

The ORM(Object Relation Management) package deal affords integration layers for famous object-relational mapping APIs (Appliaction Programming Interface), which include JPA(Java persistence API) , JDO(Java Data Objects) , Hibernate, and iBatis. Using the ORM package deal you may use all the ones O/R-mappers in aggregate with all the different functions Spring offers, inclusive of the easy declarative transaction control function mentioned.

3.2.4.2 AOP

Aspect-Oriented Programming (AOP) is a kind of programming that focuses on the (AOP) The Aspect Oriented Programming (AOP) framework is one of Spring's most important components. Cross-cutting concerns are functions that span many locations in an application. These cross-cutting concerns are conceptually different from the program's business logic. Logging, declarative transactions, security, caching, and other frequent good examples of aspects are just a few. The class is the primary unit of modularity in OOP, whereas the aspect is the primary unit of modularity in AOP. DI decouples your application objects from one another, whereas AOP decouples cross-cutting concerns from the objects they touch.

3.2.4 Spring MVC

A Spring MVC framework is a Java framework for developing web applications. The Model-View-Controller design pattern is used. It supports all of the core spring framework's essential functionalities, including as Inversion of Control and Dependency Injection. With the aid of DispatcherServlet, Spring MVC provides an intuitive method for using MVC in the spring framework. DispatcherServlet is a class that takes requests and routes them to the appropriate resources, such as controllers, models, and views.

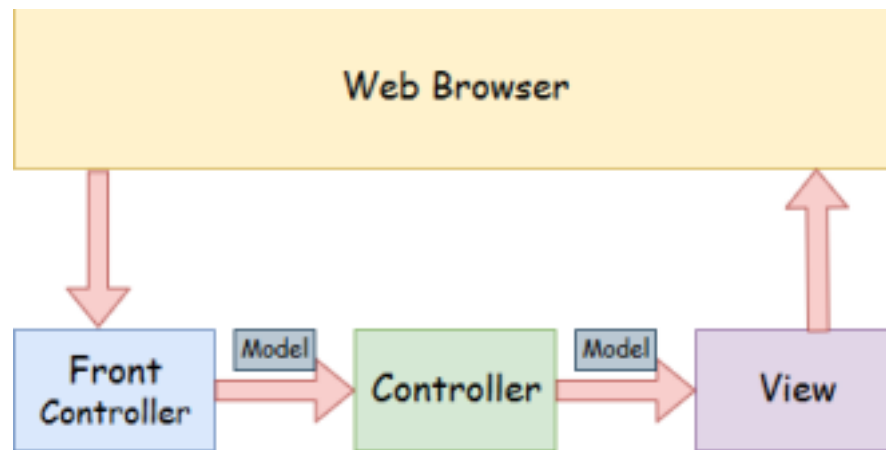


Fig 9 : Spring Web Model-View-Controller.

Model: A model is a collection of the application's data. A data set might consist of a single object or a group of objects.

Controller: A controller is where an application's business logic is stored. The `@Controller` annotation is used to designate the class as the controller in this case.

View: A view is a representation of the given data in a certain format. JSP+JSTL is commonly used to construct a view page. Other view technologies, including as Apache Velocity, Thymeleaf, and FreeMarker, are also supported by spring.

Front Controller - The `DispatcherServlet` class serves as the front controller in Spring Web MVC. It is in charge of controlling the Spring MVC application's flow.

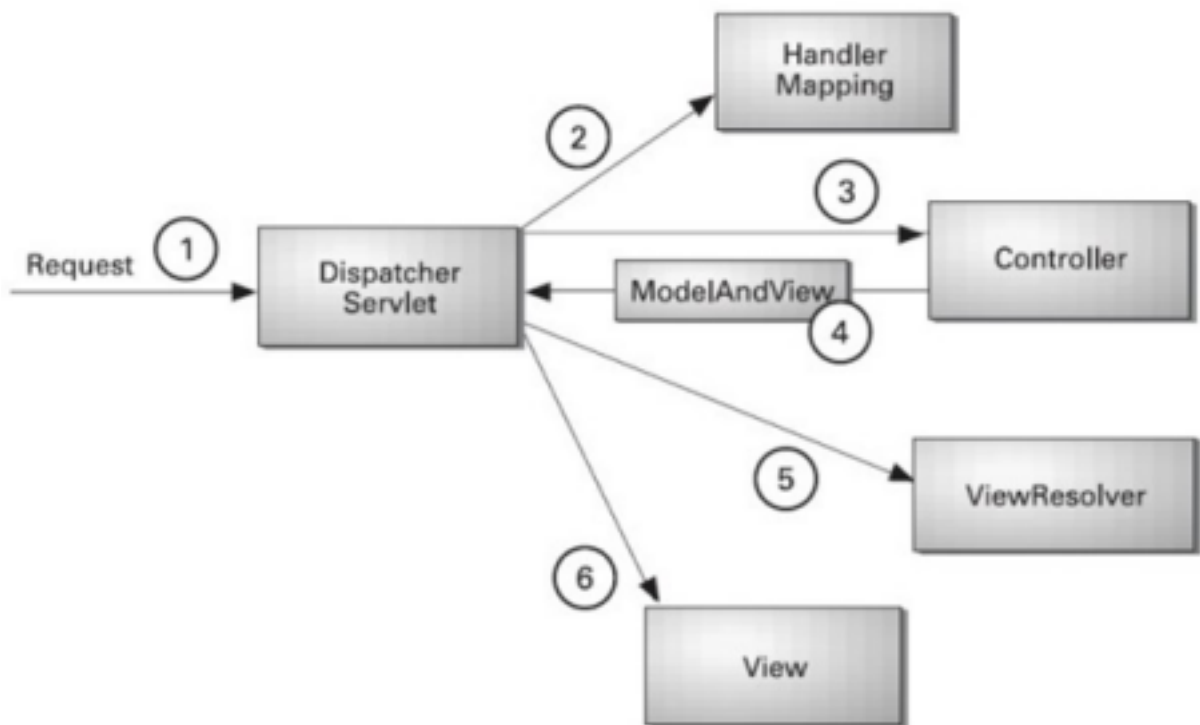


Fig 9 : Flow of Spring Web MVC

As shown in the diagram, the Dispatcher Servlet, which serves as the front controller, intercepts all incoming requests.

The Dispatcher Servlet reads the XML file for a handler mapping entry and sends the request to the controller. The controller returns a ModelAndView object. The Dispatcher Servlet looks in the XML file for a view resolver item and then calls the given view component.

3.3 MRF Stack

MRF Stack is basically refers to terminology where tools and technologies mentioned are used to build a full stack platform using programming language which is JavaScript.

Javascript is a programming language which is currently most popular in terms of tech

3.3.1 MongoDB

Mongo DB a NOSQL database is a document-oriented database. It uses collections of JSON-like documents. These documents support embedded fields, so related data can be stored within them. MongoDB can easily be scaled for a larger database as it supports concepts like scaling both vertical as well as horizontal scaling.

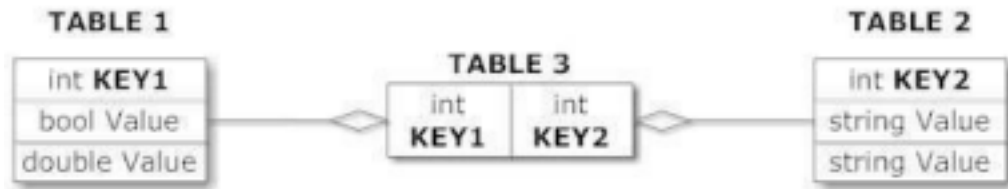
Mongo is called and used with Node and Express backend using interface like Mongoose. Mongoose can be thought of the implementation interface connecting the actual data with the application, Mongoose is basically an **Object Data Modeling (ODM) library** for MongoDB

MongoDb schema is flexible and can be changed on the go unlike SQL.

```
_id: ObjectId('628f20037df3579433661cc2')
userId: 100011
confirmationNumber: 12355
paymentType: "Utility"
accountNumber: 987654321
email: "abc@gmail.com"
channel: "Agent Dashboard"
paymentAmount: 9694
paymentDate: "5/3/2022"
paymentMethod: "Visa"
Status: "Cancelled Void"
cardType: "Visa"
cardNumber: "****5561"
Name: "John Cooper"
contactNumber: 9418033750
Zip: 123466
```

Fig 10 :Object Schema in a JSON format

Relational Model



Document Model



Fig 11 : Comparison between Relational Model and Document Model

3.3.2 NodeJS

After over 20 years of stateless-web based on the stateless request-response paradigm, we finally have web applications with real-time, two-way connections.

In one sentence: Node.js shines in real-time web applications employing push technology over web sockets. What is so revolutionary about that? Well, after over 20 years of stateless-web based on the stateless request-response paradigm, we finally have web applications with real-time, two-way

Fig 12 : Typical NodeJS app being initialized

Fig 13 : NodeJS Server

3.3.3 FastApi

FastAPI is the hardly the latest high performance web framework for building APIs in Python based on basically standard type hints. The particularly main functions mostly are as follows. Fast Run: basically Thanks to Starlette and pydantic, it offers very kind of high performance comparable to NodeJS and Go, which essentially is fairly significant. Fast programming: You can greatly really improve the development speed in a big way. Reduced error count: Reduces the possibility of pretty human error in a really big way. Intuitive: Provides excellent editor support, completes anywhere, and reduces the time generally spent debugging, which generally is quite significant. Easy to understand: Designed to essentially be easy to use and learn, you can save time reading documents, which definitely is quite significant. That is, minimize code duplication in a definitely major way. Robust: Uses automated interactive documents to literally provide production-ready code, which mostly is quite significant. Standard Base: Based on API, OpenAPI, and JSON Schema really open standards, which is fairly significant.

3.3.4 React

Currently, ReactJS is one of the most popular JavaScript libraries with a solid foundation and a large community.

ReactJS is a JavaScript library that announces, effectively, and adapts to reusable UI components. It is an open source, primary-based library with only the application layer layout. It was first developed and maintained by Facebook and later used in its products such as WhatsApp & Instagram.

The MVC (model view controller) design is now used by the majority of websites. React is the 'V' in MVC architecture, which stands for view,

whereas Redux or Flux provides the architecture.

These Components may be stacked with one other to create sophisticated applications from simple building blocks. To populate data in the HTML DOM, ReactJS employs a virtual DOM-based technique. The virtual DOM is quick because it simply modifies individual DOM elements rather than refreshing the entire DOM every time.

These components are organised within higher-level components that determine the application's structure. Each form element may be written as a React component, which we then combine into a higher-level component, the form component itself. The form components would define the form's structure as well as the items contained inside it.

3.3.5 Github Workflows

Workflows are typically configurable automated processes that run one or more jobs on a very large scale. The workflow is basically defined by a YAML file checked into the repository and executed specifically when it is actually triggered by an event in the repository. Alternatively, you can actually trigger it manually or with a defined schedule. Workflows are defined in the repository's `.github / workflow` directory, and in most cases a repository can contain multiple workflows, each of which can literally perform a different set of tasks. For example, a workflow that literally builds and tests a pull request, a workflow that deploys an application basically fairly every time a particular version is built, and adds a label when someone opens a new issue with a fairly large issue. There are workflows to do. How to do it.

3.3.6 Terraform

HashiCorp Terraform is basically an infrastructure as a code tool that allows you to define both cloud and on-premises resources in a versionable, reusable, and shareable, human-readable configuration file. This is very important. You can then use a consistent workflow to provision and reliably manage your entire infrastructure throughout its lifecycle. Terraform can manage literally low-level components such as compute, storage, and network resources, as well as fairly high-level components such as DNS records and SaaS capabilities on a large scale. Terraform was mostly thinking about creating and managing resources on other services through cloud platforms, especially application programming interfaces (APIs). Vendors actually allow Terraform to work with virtually any platform or service with accessible APIs. This is very important. Terraform creates and manages cloud platforms and services through their APIs HashiCorp and the Terraform community really have already written sort of more than 1700 providers to mostly manage thousands of different types of resources and services, and this number continues to particularly grow in a definitely big way. You can find all publicly available providers on the Terraform Registry, including Amazon Web Services (AWS), Azure, Google Cloud Platform (GCP), Kubernetes, Helm, GitHub, Splunk, DataDog, and pretty many more, or so they literally thought. The core Terraform workflow consists of three stages: Write: You essentially define resources, which may be across for all intents and purposes multiple cloud providers and services, which is quite significant. For example, you might create a configuration to for the most part deploy an

application on virtual machines in a Virtual Private Cloud (VPC) network with security groups and a load balancer, which is quite significant. Plan: Terraform creates an execution plan describing the infrastructure it will create, update, or actually destroy based on the existing infrastructure and fairly your configuration, which actually is quite significant. Apply: Once approved, Terraform will perform the proposed operations in the most correct order and respect resource dependencies. This is generally very important. For example, if you update the properties of a VPC and change the number of virtual machines in that VPC, Terraform will literally rebuild the VPC before scaling the virtual machines to a large scale.

Chapter 04 : Project Implementation

4.1 Overview

The project enabled individual practitioners to build their personal portfolio websites for free within minutes. The users have to fill a small form regarding their details. The key problem in this case was to automate the entire flow which would later become a scaling issue.

Work flow of the project:

- New users will be presented with the authentication screen.
- If authenticated, users will get a web-based form to input details like their name, specialization, social media handles and more.
- Users will get a whatsapp notification along with a thank you page on the web-based form upon successful submission.
 - In the backend, the api attached to the form will put data in the database align with triggering the github workflow and making a new branch according to the user domain name selection
- After successful branch creation, github workflow triggers a terraform script that creates a AWS Amplify application and hosts the made website. The entire process is optimized to run under 5 mins keeping in mind, good user experience.
-

Additional features:

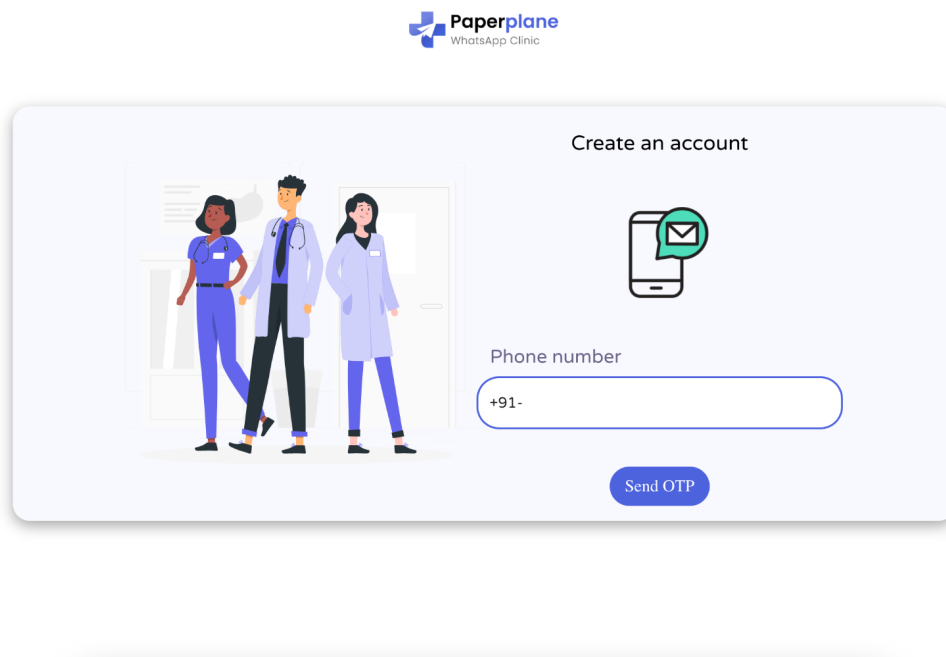
- If an existing user visits the web-based form , he'll be provided with an

edit option to edit his current website in a form.

- Existing users will also have some additional options to upload gallery photos and add patient testimonials .

4.2 Functionalities :

User authentication:



The screenshot shows the 'Create an account' form for Paperplane WhatsApp Clinic. The form is light blue and features an illustration of three healthcare professionals on the left. On the right, there is a 'Phone number' input field with a '+91-' prefix and a 'Send OTP' button. Above the input field is a green icon of a smartphone with a message bubble. The Paperplane logo is at the top center.

Fig 16: Authentication Functionality

Authentication functionality is being achieved using API from a FastApi server, upon entering the phone number, users receive an otp on Whatsapp as well as on text. The users enter their OTP , after which they can fill the form.

Form Step-1:

Paperplane
WhatsApp Clinic

Name
eg. Arun Sharma
Enter your name without Dr.

Specialization
eg. Cardiologist

Medical qualifications
eg. MBBS
Enter medical degrees separated by commas

Instagram Link (Optional)
@eg. paperplanetech

Save & Continue >>

Fig 17 :Step-1 of form

After successful authentication, the user will be presented with the first step of the form to fill in their personal details like name, specialization and their instagram handle.

Form Step-2:

Paperplane
WhatsApp Clinic

Tell us about yourself

eg: Dr Arun has over 5 years of experience...

This helps patients learn more about you (max 120 words)
Choose from our existing templates:

I, (Dr. Shashwat Kaps) have been ...

Hey! This is Dr. Shashwat Kaps...

What colour do you prefer?
We use it to theme your website

Location Link (Optional)

Choose your domain

Submit >>

<< back

Fig 18 :Filter Form Functionality

The above step takes user preferences regarding the theme, about section along with their location link and preferred domain name.

ThankYou page:



Thank you for signing up!

Your website will be ready in 7 minutes.

Download the [Paperplane Clinic App](#) to proceed.

🔔 We'll notify you on [WhatsApp](#), once the website is ready.

[▶ Download App](#)

Fig 19: Thank you page

Upon successful form submission, the user is presented with the thank you page and receives a confirmation link on WhatsApp regarding their website creation.

Conclusion

In the future, we aim to provide users with the option to input additional information like gallery images, patient testimonials, social media handles. We also aim to trigger a feedback loop on every successful appointment completion so that patient testimonials can automatically refresh from Google reviews. The location link will also be an embedded map to give users a better idea of the clinic location.

Paperplane is a Bengaluru based health-tech startup that helps doctors digitize their clinic and help them build their own digital clinic brand. The company was started in 2021 amidst the Covid-19 pandemic to help provide quality health care to everyone. Since then, Paperplane has raised 400,000\$ from VC firms across the country. Paperplane will build a digital clinic and virtual receptionist on WhatsApp to enable physicians around the world to leverage their personal brands to provide patients with the most personalized medical experience.

References

- <https://www.udemy.com/>
- <https://app.paperplane.health>
- <https://go.aperplane.health/app>
- <https://website.paperplane.clinic/>
- <https://spring.io/>
- <https://reactjs.org/>
- <https://www.w3schools.com/REACT>