SOFTWARE ENGINEER – INTERN

PAXCOM, MOHALI, PUNJAB

DATE: 7TH FEBRUARY 2022- PRESENT

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

Bachelor of Technology

In

Computer Science and Engineering

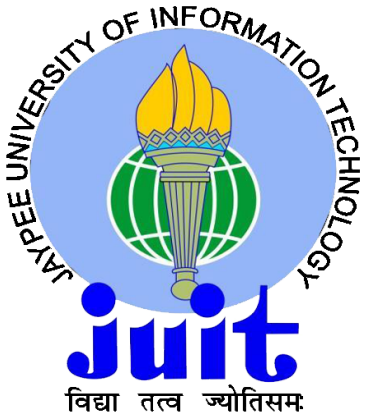
By: Archit Sharma

181276

Under the supervision of

Mr. Pankaj Chauhan, Mr. Shikshank Sharma

Dr. Yugal Kumar



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 **"Software Engineer – Intern – Paxcom Pvt Ltd. Mohali, Punjab"** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat, is an authentic record of work carried out by **Archit Sharma** 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 Archit Sharma and Paxcom Pvt Ltd. Mohali, Punjab, dated 10 February, 2022.

Archit Sharma
181276
Jaypee University of Information Technology Waknaghat, Solan, H.P.

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

Mr. Pankaj Chauhan (Sr. Software Engineer)

Mr. Shikshank Sharma (Sr. Software Engineer)

Paxcom Pvt Ltd. Mohali, Punjab.

Dr.Yugal Kumar (Assistant Professor)
Jaypee University of Information Technology Waknaghat, Solan, H.P

ACKNOLEDGEMENT

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 Paxocm 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 institution, but will also be useful for all those who are interested to learn about internship experiences in auditing and consulting firm.

Archit Sharma
181276
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 under as Software Engineer - Intern at Paxcom 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 giving their details in the references.

Archit Sharma
181276
Jaypee University of Information Technology
Waknaghat, Solan, H.P

INTERNSHIP REPORT UNDERTAKING

I, Archit Sharma, Roll No. 181276, Branch – Computer Science and Engineering currently pursuing my internship with Paxcom 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, but due to COVID-19 situation and Work from Home procedure being followed, 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 Training and Placement Cell whenever COVID-19 situation gets normal.

Archit Sharma
181276
Jaypee University of Information Technology
Waknaghat, Solan, H.P

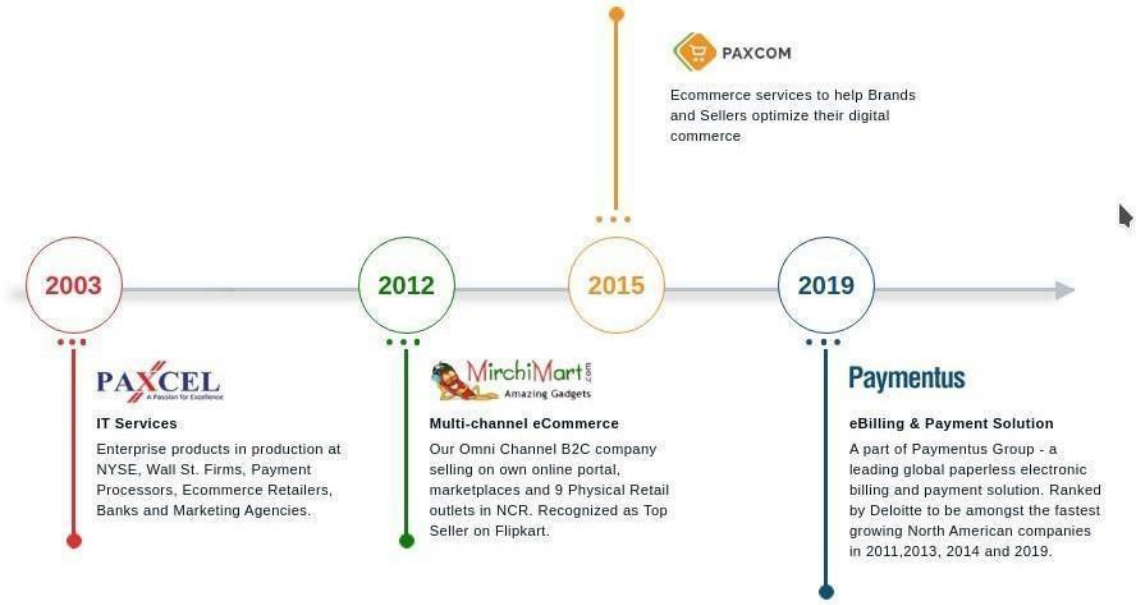
TABLE OF CONTENT

Chapter 01 : INTRODUCTION

1.1 Introduction

Paxcom

Paxcom is a team of over 400 e-commerce enthusiasts dedicated to using technology to simplify digital commerce and payments for brands around the world. Paxcom is part of the Paymentus Group, a world leader in paperless electronic payments and payment solutions. Paxcom`s e-commerce solutions include analytics software, advanced data analytics, marketing and advertising, content, sentiment analytics, web store building, ordering and inventory management software. Their state-of-the-art solutions collect and analyze data from all e-commerce channels for useful insights, inventory rates, availability, competitive pricing, product discoverability, and marketing costs, It also offers improved ROI, placement, etc. Paxcom has 148 platforms, 450+ PIN codes, 1.4+ rack products, multi-crawl advertising spending per day, and seller and location-specific to drive e-commerce and gain customer support. It is part of an multi company group that researches and works in various industries such as ecommerce, IT services, billing and payment solutions. Paxcom is a unique SAAS-based software that provides a complete e-commerce automation solution from e-store and marketplace creation and management to advanced data analytics, rapid sales growth, managed warehouses and supply chains. It also improves customer satisfaction.



Paymentus

Paymentus is a North Carolina based software company providing complete billing solutions. Paymentus was basically created with the desire to simplify and change how the invoices are paid and improve it using innovation and technology. With drive to innovative and simply the complex problems using technology, Paymentus has grown to be the market leader in sectors of paperless electronic billing and other related payment solutions with 1,700 customers it includes the largest claimant in North America.

Paymentus is recognized by Deloitte as one of the fastest growing North American companies in the four years, Paymentus is continuously working to develop billing and payment platforms that are better, faster, safer and more cost-effective. Paymetntus is a highly customer driven company and works greatly to improve customer satisfaction and work contiusoly to

provide best to its cliental.

This internship was with Paxcom India, The project / team which was allotted was for Paymentus India Pvt Ltd at Mohali branch. The internship was with the Team working on payment dashboarding development and related products along side of other payment data tasks mostly based on Java Core as well as Java Frameworks like Struts2, JDBC(Java Database Connectivity API), Spring MVC(Model, View, Control) as well as SpringBoot alongside extensive taks on MERN(MongoDB, Express, React, NodeJS) tech stack, the major focus was on successfully understand their working structure and pattern along side gaining extensive skill set as Full Stack Engineer and related underlying technologies. focus was on learning soft skills like communicating within a corporate firm and working with team.

Internship goals included successfully understand 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. Paymentus has a typical blend of work and fun. Internship included 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 unser 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 most

stunning 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 projects and get appreciated for it. The way that this entry-level position straight forwardly identified with what was required to get started on company projects from contributing to actually owning them was of immense help. The internship at Paxcom, paymentus was a great opportunity to learn, test and work on Full Stack Engineering skills along side working on soft skills like communication.

Chapter 02 : DESCRIPTION OF THE INDUSTRY

2.1 FinTech (Financial technology)

The term FinTech is made up two words Finance 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 the finance.

From the ability for consumers or users to view and make financial transactions online, to applications that allows them to make financial transaction within seconds to anyone across the globe, to means that enable financial institutions to make rapid financial transactions in huge volume and velocity, everything is the sub domain of innovation in finance aided with technology. FinTech's closest examples in our day to day life would be implementation of apps like Paytm, GPay etc. or the investment apps like Grow, Zerodha.

One of the biggest opportunity provided by the advent of the fintech in the current marketspace is in terms of aiding the common man with the financial literacy as well as financial accessibility, it has brought more and more control to people’s life in terms of their financial decisions and investments, it has opened the investment opportunities which were earlier beyond the reach of common man.

FinTech's growth is primarily due to the opportunity for smaller players to compete in the same space as traditional banks and financial institutions. Thanks to FinTech, it's not about who is the biggest, but who is the fastest and most responsive to effectively respond to the ever-changing consumer

demands. Moreover, the solutions offered by fintech companies are no longer "universal." Instead, it provides targeted (often niche) services that fill gaps in specific financial needs. It may also be offered at a much lower cost than the services offered by traditional financial providers.

2.1.1 Electronic billing

Electronic billing or eBilling is the process of sending and paying invoices electronically. This process allows customers to receive invoices in email, web portal, or machine-readable data format for more efficient delivery and payment. The eBilling process is managed by accounts receivable by receiving results from accounting software or an ERP system and providing them to customers. Aspects of this paperless process are becoming more and more automated to improve speed and accuracy, such as follow-up when invoice payment times are delayed.

With the localisation of the technology and more and more people accessing internet and other means of technology it has brought more and more people paying their bills online genratinh thousands of transactions per second and henceforth resulting in millions of invoices generated which needed to transferred, repealed and stored.

This industry revolutionizes traditional billing system in a lot of intuitive ways. It eliminates process inefficiencies, save time and money, ensure more accurate and faster delivery of invoices to your customers, and enable instant payments. A well-deployed electronic billing system is a strategic advantage, not just an operational upgrade.

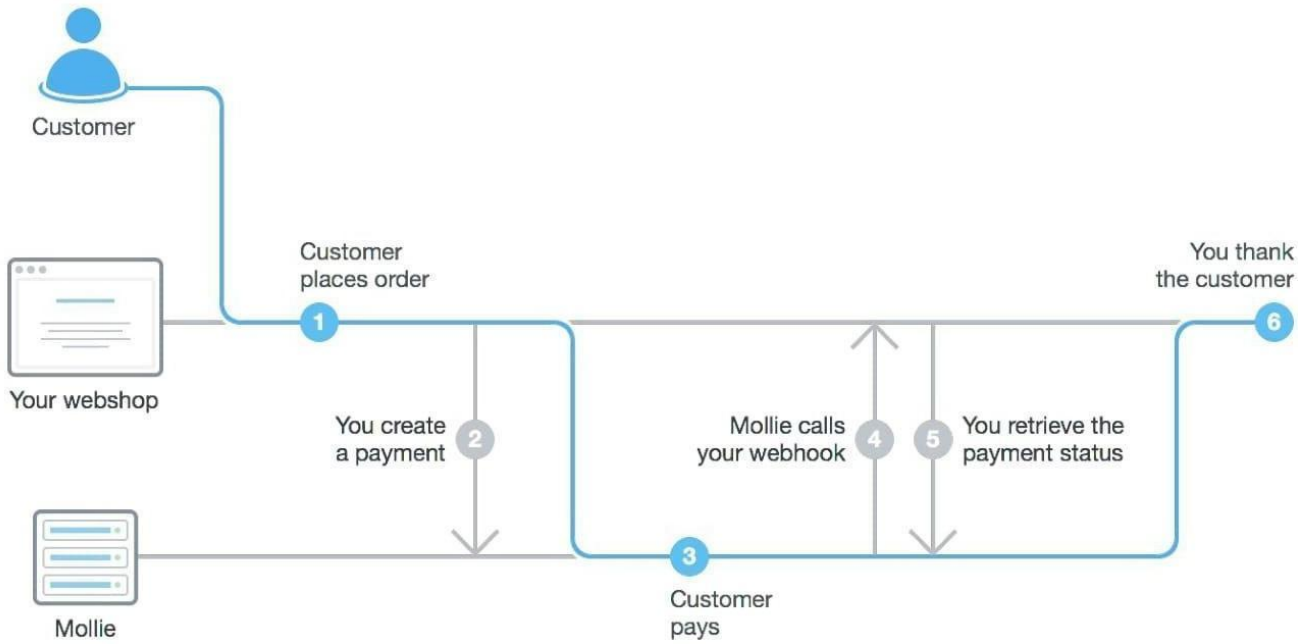
How Paymentus Works:

How to Pay:

Input the URL of the organization you want to pay into your web browser and navigate to the billing section to begin the payment process.

Unsure of the URL?:

It is often found on your paper billing statement. If not, look-up the webpage via your preferred search engine (Google, Bing, etc.).



2.2 Paymentus - Payment Dashboarding Products

Paymentus offers multiple products and services in domains of the payment be it in terms of eBilling, invoice generation or simple payment gateway kind of services.

The current team under which this particular internship is working onthe dashboading product of the paymentus . The following products are a more of a visualization, analytics as well as data querying code allowing the opportunity to the end user be it big businesses with large volume of clientele and volume and velocity of transaction or be it a single end common user to basically access, analyze the transaction done from their account and address financial questions or in case of some special test cases also offer to answer certain business logics.

Following are the major dashboards which are currently operating :-

1. **Agent Dashboard Systems** : This feature provides our clients having dedicated customer support to query their customer problems by a dedicated dashboard which includes payment reports , history visualization etc. It also provides customer support with features for IVR and Assisted IVR payments.
2. **Customer Portal** : This is the feature acts as payment solution to customers by providing them with features like wallets or direct payment portal for registered users. It is more or less acts as a direct common users all payment kind of solution and to provide them with multiple accesbilities

3. **ROTP (Remote One Time Payment)** : This feature is basically a payment gateway kind of solution for the guest users to make payments using multiple sources and multiple types. Works on similar ground as a payment gateway we see while we make payments across any given platform online.

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 Java based frameworks like Spring MVC, Spring Boot, Struts2, JDBC, Log4j2 along side working on Web Technologies like MERN (MongoDB, Express, React, NodeJS) Stack with HTML, CSS, GithHub.

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 task 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 grasp of good coding practices in line of the set practices of the code already established in 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 Developement
- 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.

DuplicateNumber.java

```java
package assignments.collections;

import java.util.HashSet;

/**
 *
 * The following class is purposed to find single duplicate
 * value in a given array
 *
 * @author archit.sharma
 *
 */

public class DuplicateNumber {

    //function to produce array and duplicate a random number
        public static int[] randomArrayGenrator() {

            //array declaration
            int[] randomArray = new int[101];;

            //fill array with numbers
            for(int i = 1;i <= 100; i++) {
                randomArray[i] = i;
            }

            //duplicate a random number
            double randomSelect = Math.random()*99;
            int randSelect = (int) Math.round(randomSelect);
            randomArray[randSelect+1] = randSelect;

            return randomArray;
        }
```

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.

```java
package jdbcpractice;

import java.sql.Connection;

public class JDBCtestpractice {
   static final String DB_URL = "jdbc:mysql://localhost/";
   static final String USER = "root";
   static final String PASS = "archit04";

   public static void main(String[] args) {
      // Open a connection
      try(Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
         Statement stmt = conn.createStatement();
      ) {
         String sql = "CREATE DATABASE STUDENTS";
         stmt.executeUpdate(sql);
         System.out.println("Database created successfully...");
      } catch (SQLException e) {
         e.printStackTrace();
      }
   }
}
```

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 easies 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 laungage.

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,

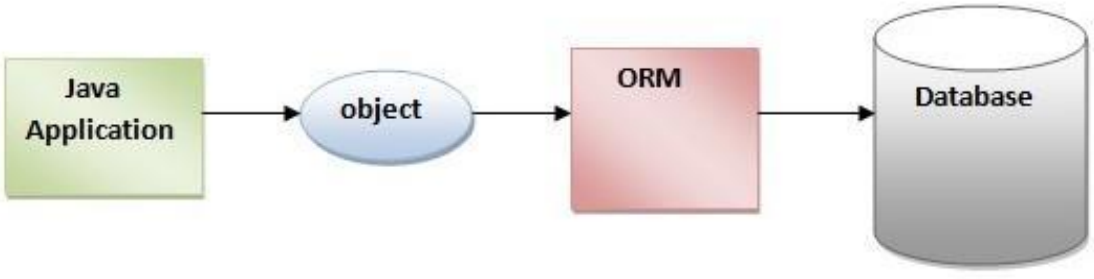


Fig 5 : Interaction with database using HIbernate layer

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/ functionalites in multiple domains. These functionalities can broadly be divided into following 6 modules:

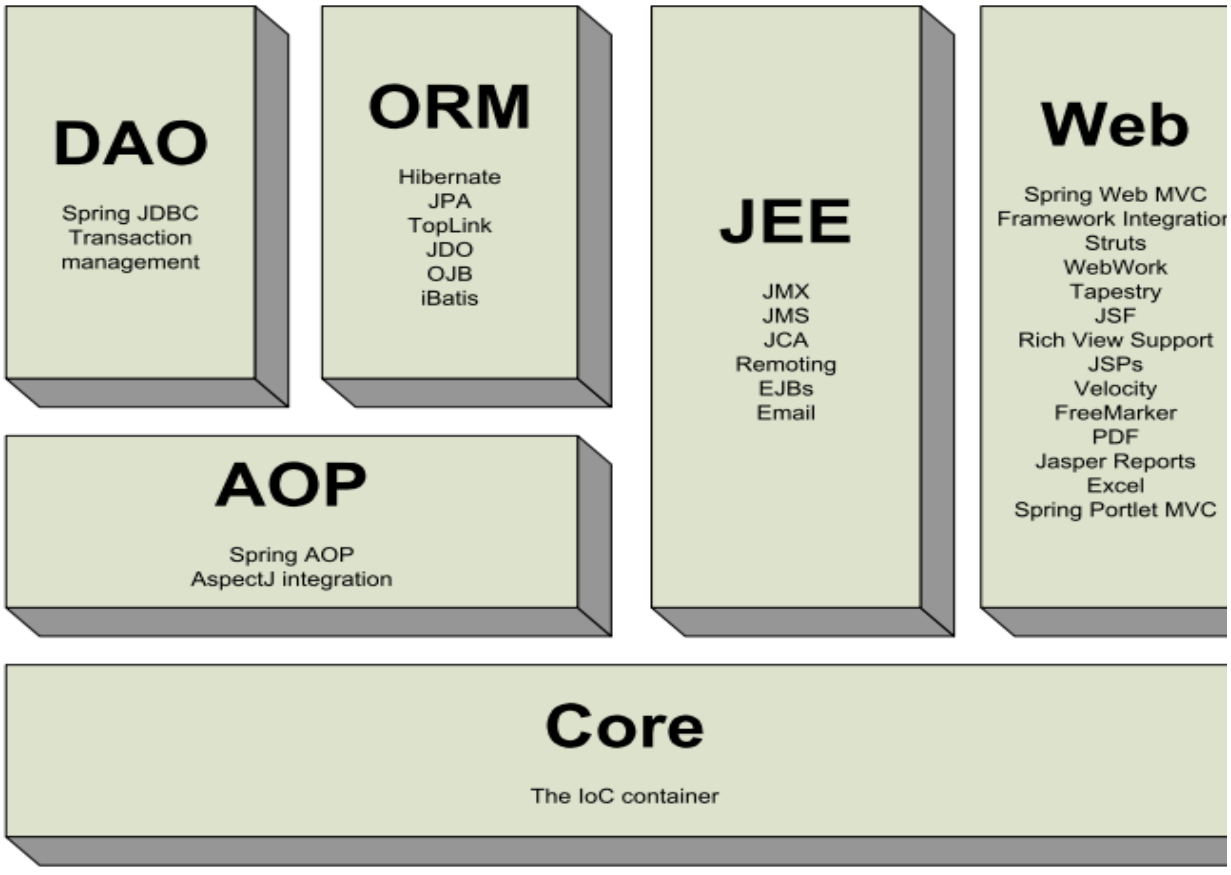


Fig 8 : Spring Framework and its modules.

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 Acess 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 Acess 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 Realation 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.

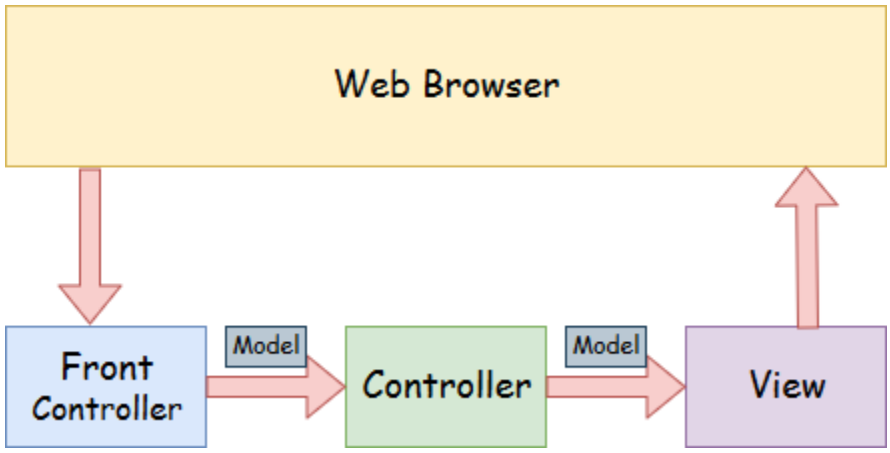


Fig 9 : Spring Web Model-View-Controller.

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.

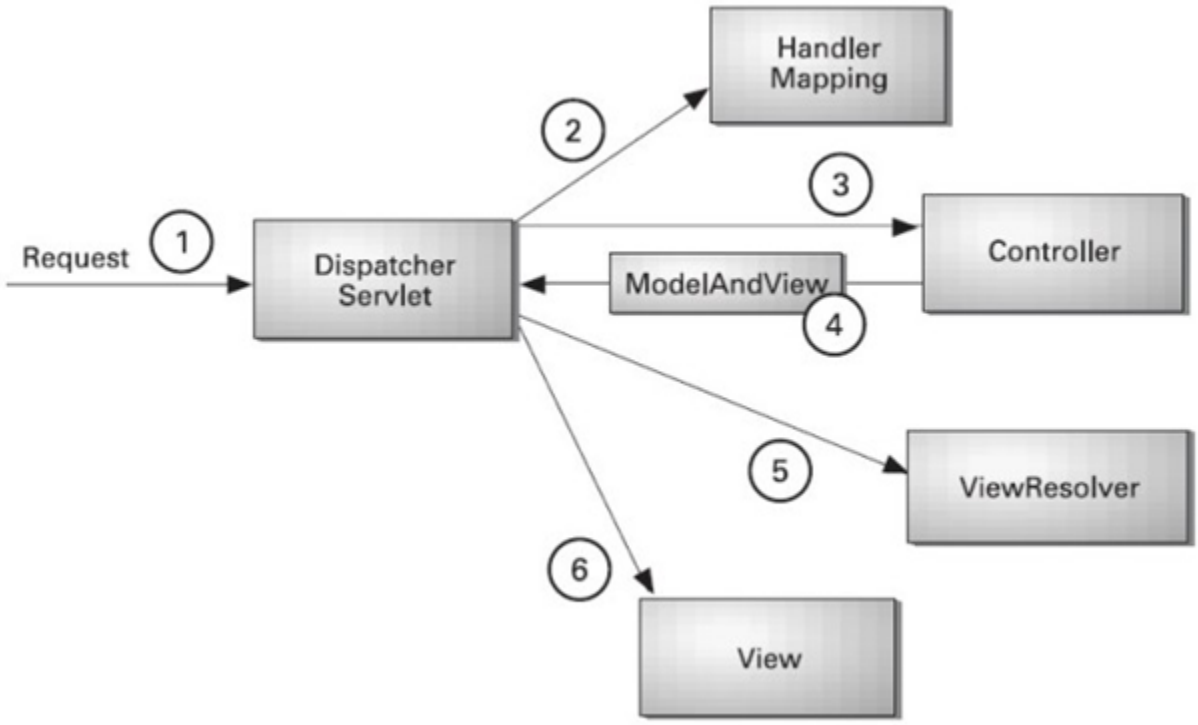


Fig 9 : Flow of Spring Web MVC

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 MERN Stack

MERN Stack is basically refers to terminology where tools and technologies mentione are used to build a full stack platform using programming laguage which is JavaScript.

Javascript is a progragramming 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-Iike documents. These documents support embedded fields, so related datacanbesto red within them. MongoDB can easily be scalled for a lager 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 Moongoose. Mongoose can be thought of the implementation interface connecting the actual data with the application, Mongoose is basically and **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

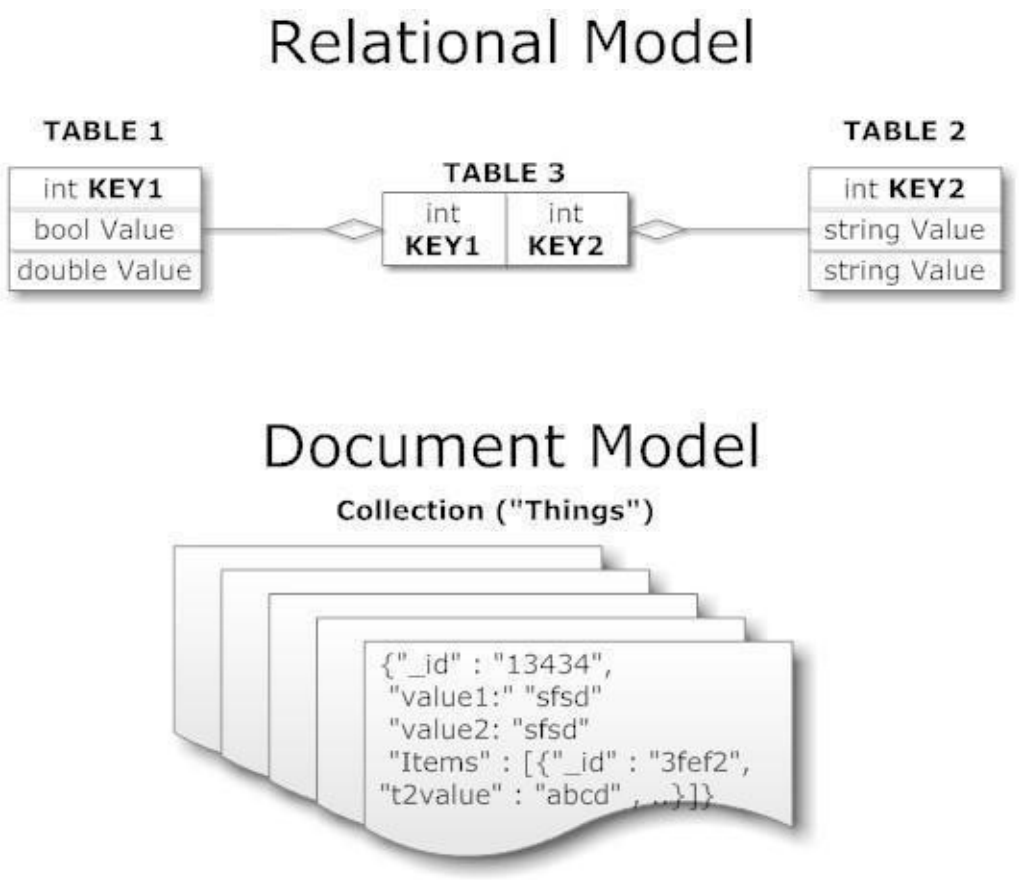


Fig 11 : Comparison between Relational Model and Dcoument Model

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? Wel, 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, where both the client and server can initiate communication, allowing them to exchange data freely. This is in stark contrast to the typical web response paradigm, where the client always initiates communication. Additionally, it's all based on the open web stack (HTML, CSS and JS) runningoverthe standard port80. One might argue that we've had this for years in the form of Flash and Java Applets -but in reality those were just sandboxed environments using the web as a transport protocol to be delivered

```
const express = require('express');
const req = require('express/lib/request');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

const cors = require('cors')



const app = express();

//cors issue
app.use(cors())

app.use(bodyParser.json());

const loginRoute = require('./routes/login');

app.use('/login', loginRoute);

const test = require('./routes/payments');
app.use('/test', test);

//

const paymentDataRoute = require('./routes/payments');
app.use('/payments', paymentDataRoute );
```

Fig 12 : Typical NodeJS app being intialized

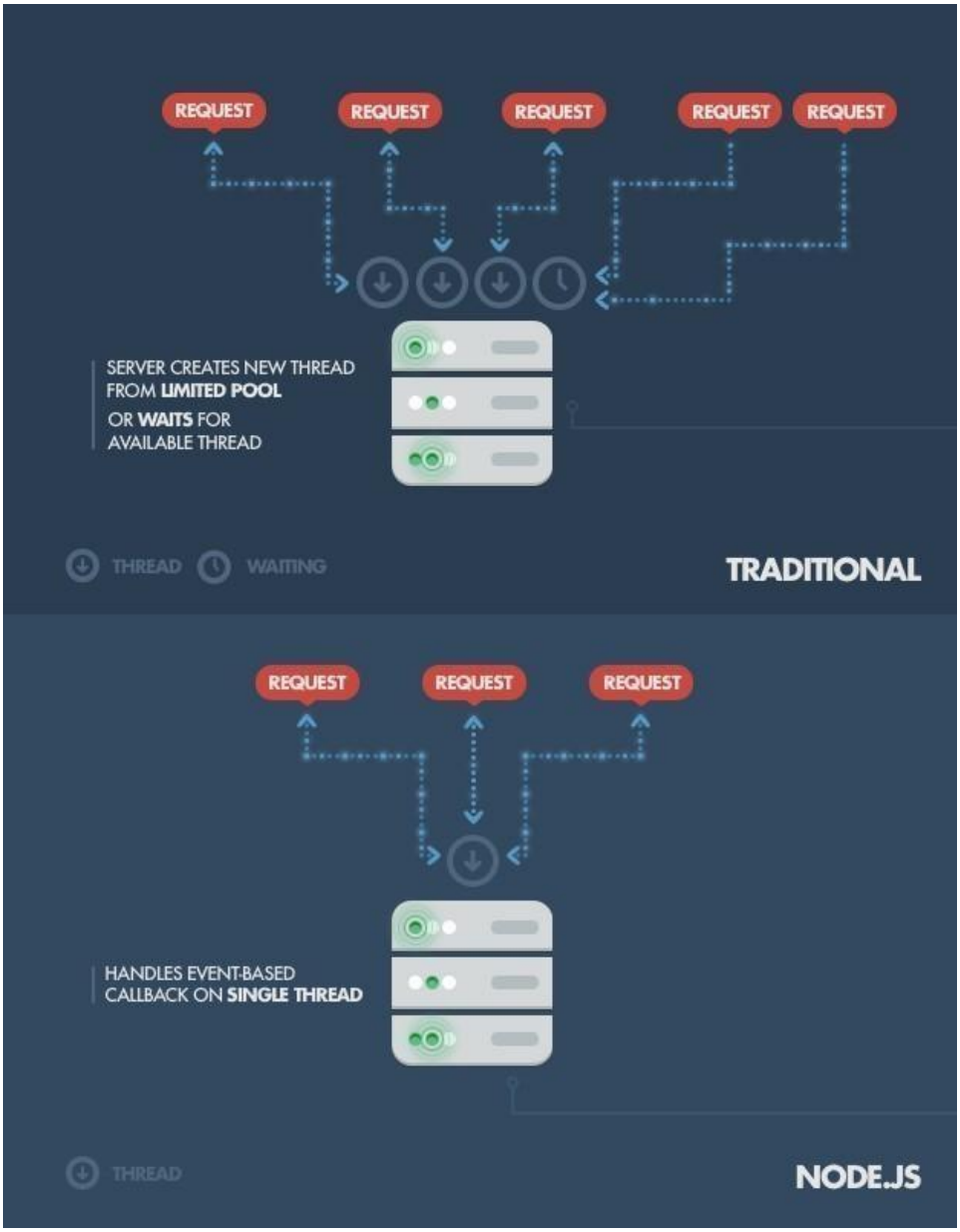


Fig 13 : NodeJS Server

3.3.3 Express

Express provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express.

```
var express = require('express');
var app = express();

app.get('/', function(req, res){
   res.send("Hello world!");
});

app.listen(3000);
```

Fig 14 : ExpressJS Code

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.

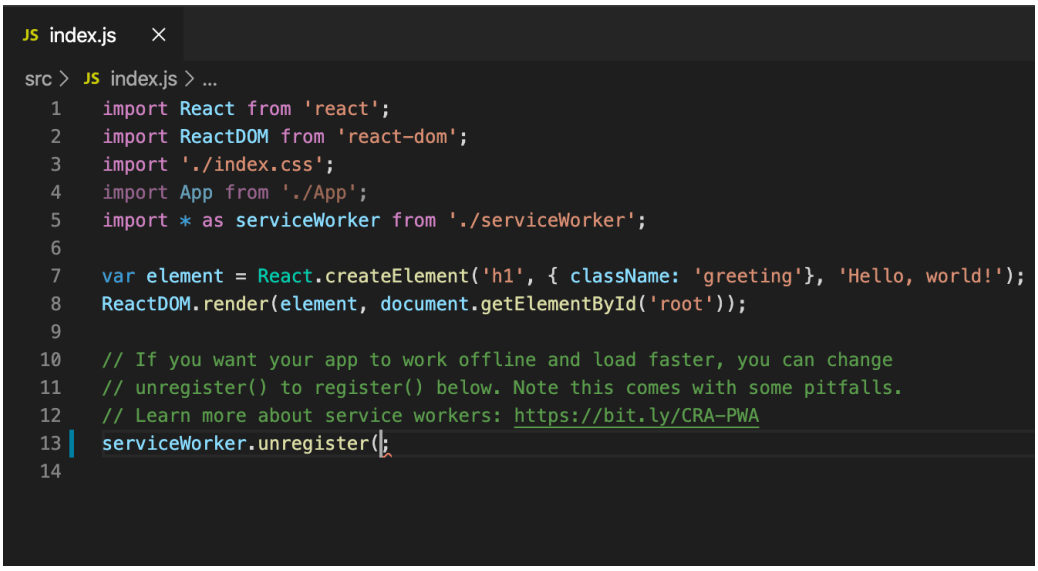


Fig 15 :ReactJS Typical Example

Chapter 04 : Project Implementation

4.1 Overview

Payments dashboard is a MERN stack project which is a part of Green Dot project being developed under Paymentus. Payments Dashboard is a web dashboard which will be used by the user to find the transaction details using the filters embedded into the dashboard. Based upon the filter, the API will fetch the transaction records from the database and display them in the table under the filter form. Each transaction will have a “Details” button, which upon clicking will show up the transaction details in another webpage.

Work flow of the project:

- Users will input credentials and based upon the validation, user will be either redirected to dashboard, if credentials are valid, or shown “Invalid Credentials” message.
- If authenticated, user will input the filter as per requirement and click on the “Search” button.
- Based upon the filters inputs by the users, API will fetch the data from the database and display the data in the table below the filter form.
- On the form, user will be able to select “No. of Records” per table.
- Users will be able to see the transaction details upon clicking the “Details” button, transaction details will show in different page.

Additional features:

- If an unauthenticated user tries to access the restricted URLs, they will be shown appropriate message.
- If user is inactive for 10 minutes, user will see a pop up asking if user want to logout or stay logged in. And if user doesn't choose, they will be logged out after 5 minutes.

4.2 Funcationalities :

Login Functionality:

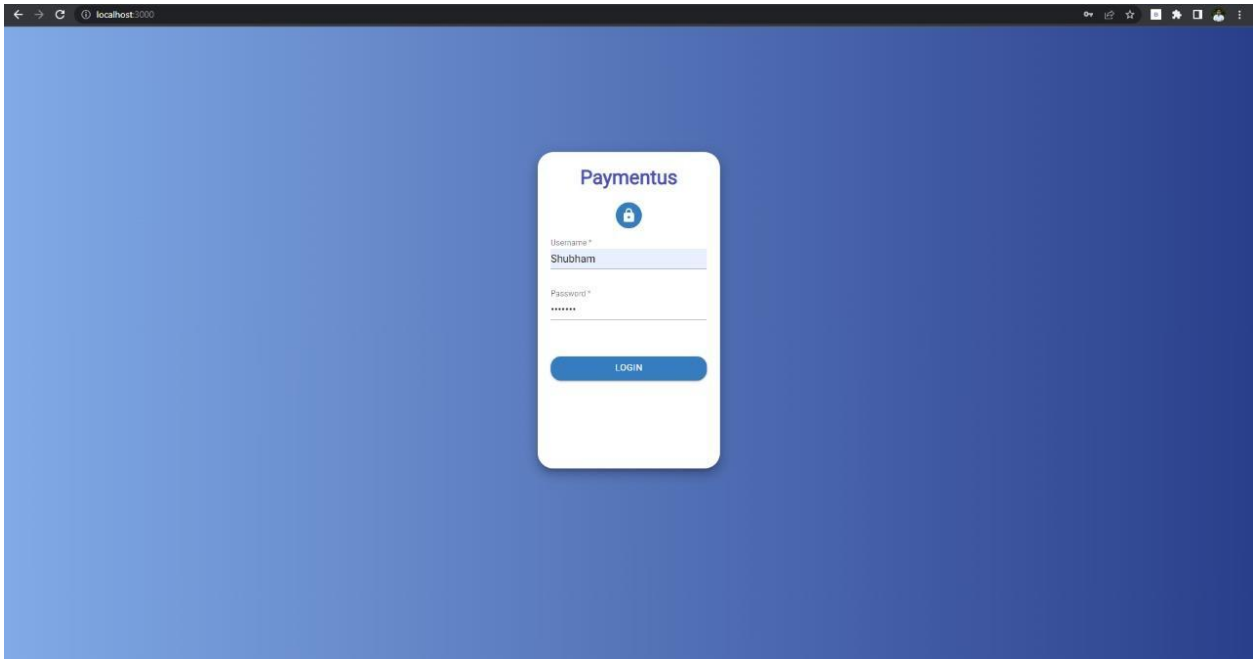


Fig 16: Login Fucntionality

Login functionality is being achieved using API from a Node.js server, upon entering the credentials, credentials will be validated, if they are valid, a token

and a refresh token will be returned otherwise error message will be returned which will be shown to the users. If credentials are valid, they will be authenticated and redirected to dashboard.

Dashboard:

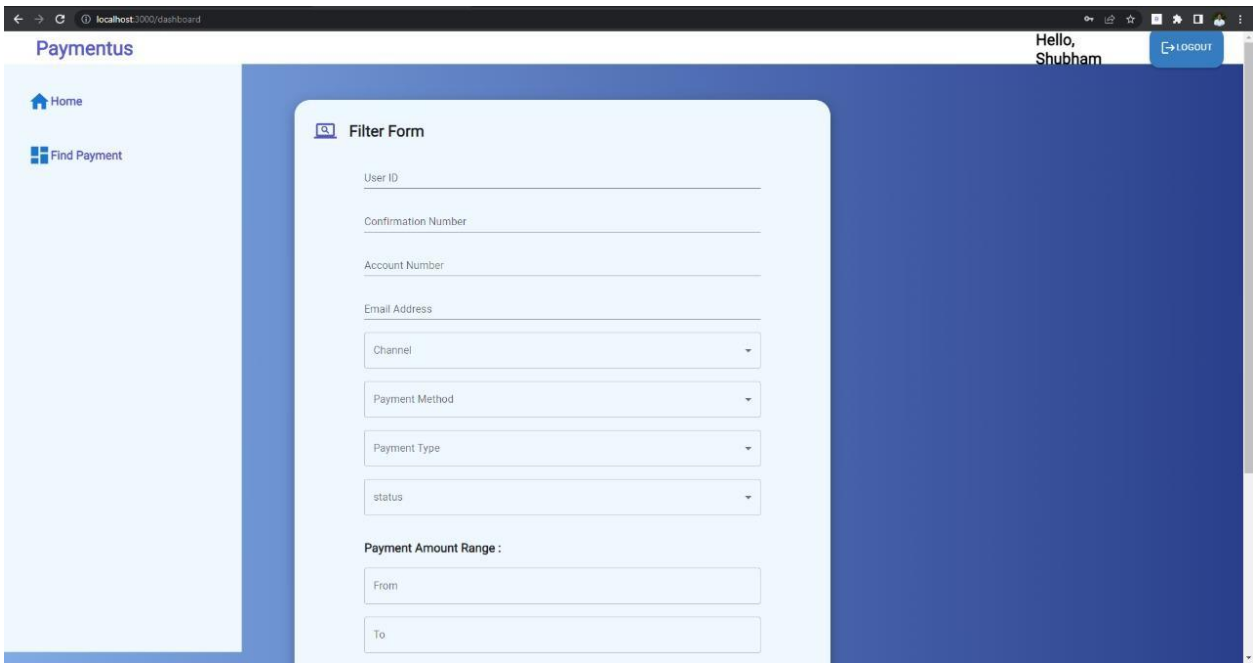


Fig 17 :Dashboard

After the user successfully logs in, they will be redirected to the dashboard where on the left they'll see a sidebar with navigation routes. On the topbar, there's a logout button. In the middle there's a Filter form which provides the main functionality of filtering and fetching the data.

Filter form functionality:

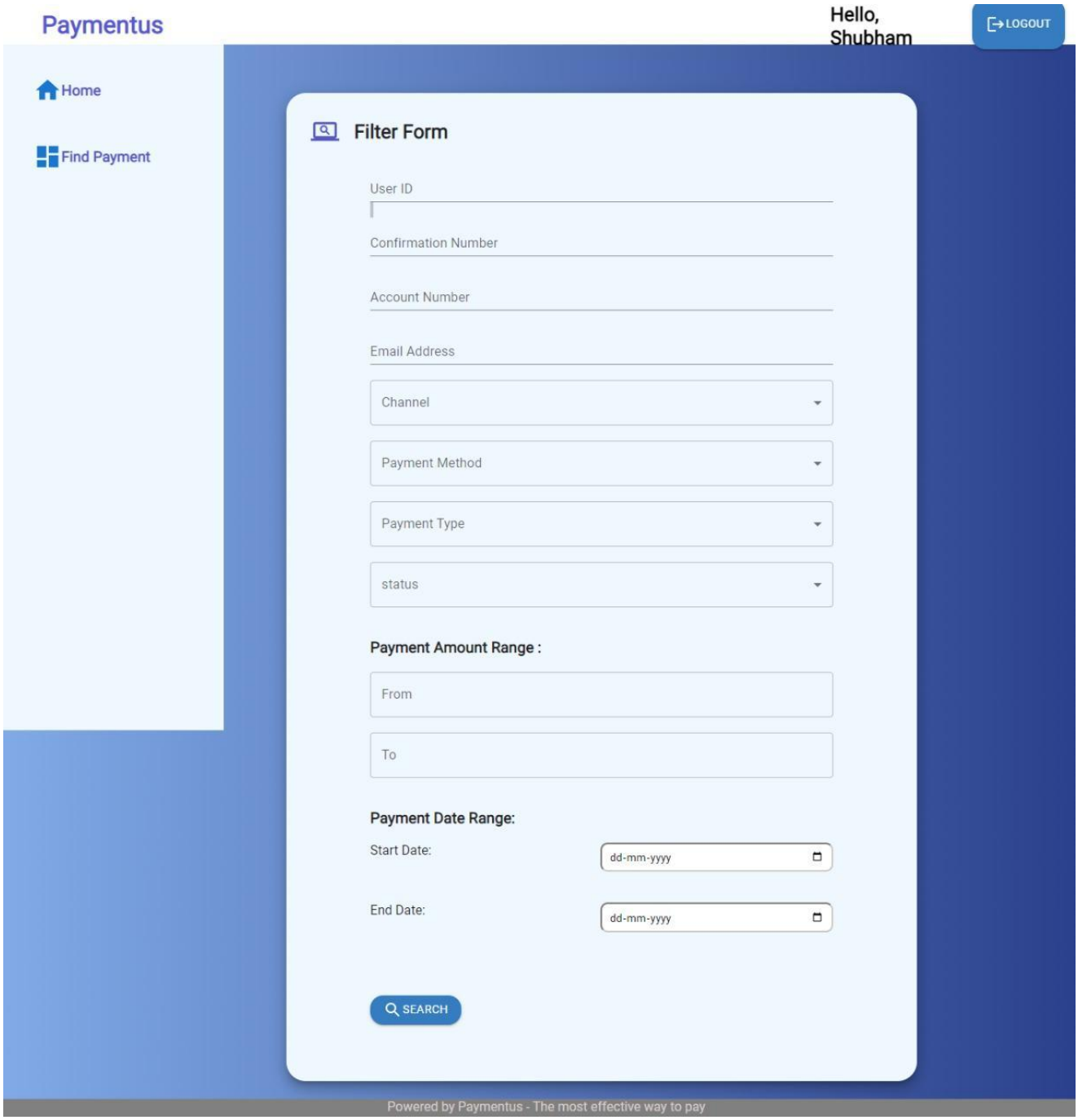


Fig 18 :Filter Form Functionality

The above form provides the main functionality of hitting the API with fields and returns the records based upon the filter.

Transaction table functionality:

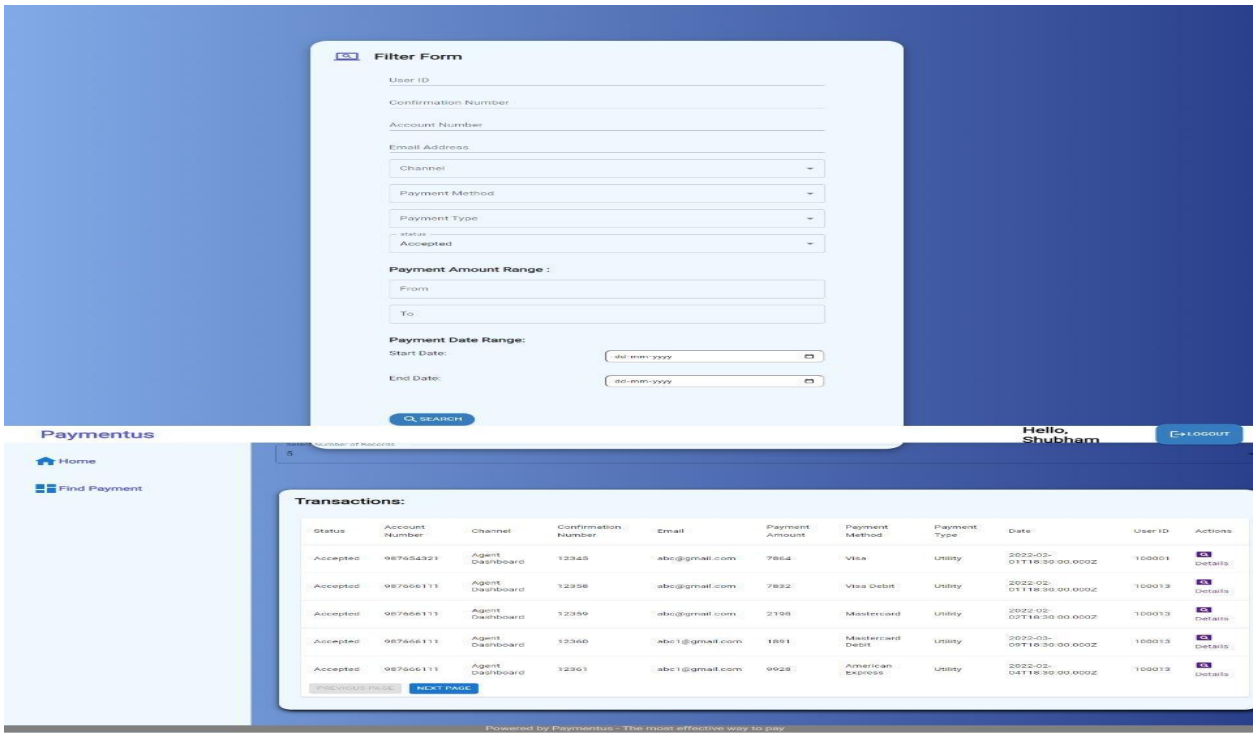


Fig 19: Transaction Table

Transaction table filled with records fetched from the API after filling the filter form. By default 5 records will be shown in the first page, user can see more by clicking the next button. Users will also have the option to choose how many records they want to see per page. Users can also check a particular transaction's details, by clicking on the "Details" button in the Actions columns.

Transaction Details functionality:

Fig 20 : Transaction Details functionality

Transaction details page will display the transaction detail of a particular transaction if a user clicks on details button for a particular transaction.

Other functionalities:

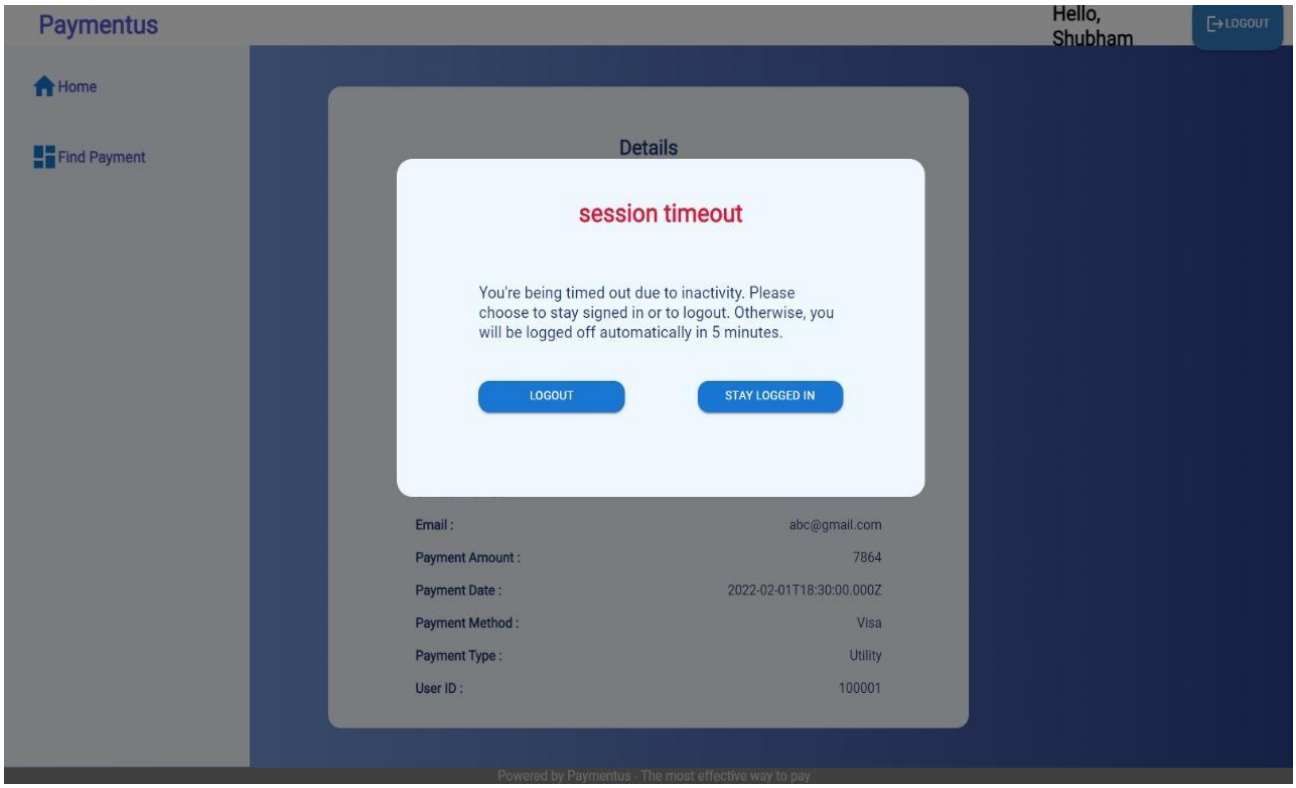


Fig 21 : Transaction Details functionality

After 10 minutes of inactivity, user will be shown above pop up modal, user will be asked if they want to stay logged in or logout. If user doesn't pick, 5 minutes later they will be logged out.

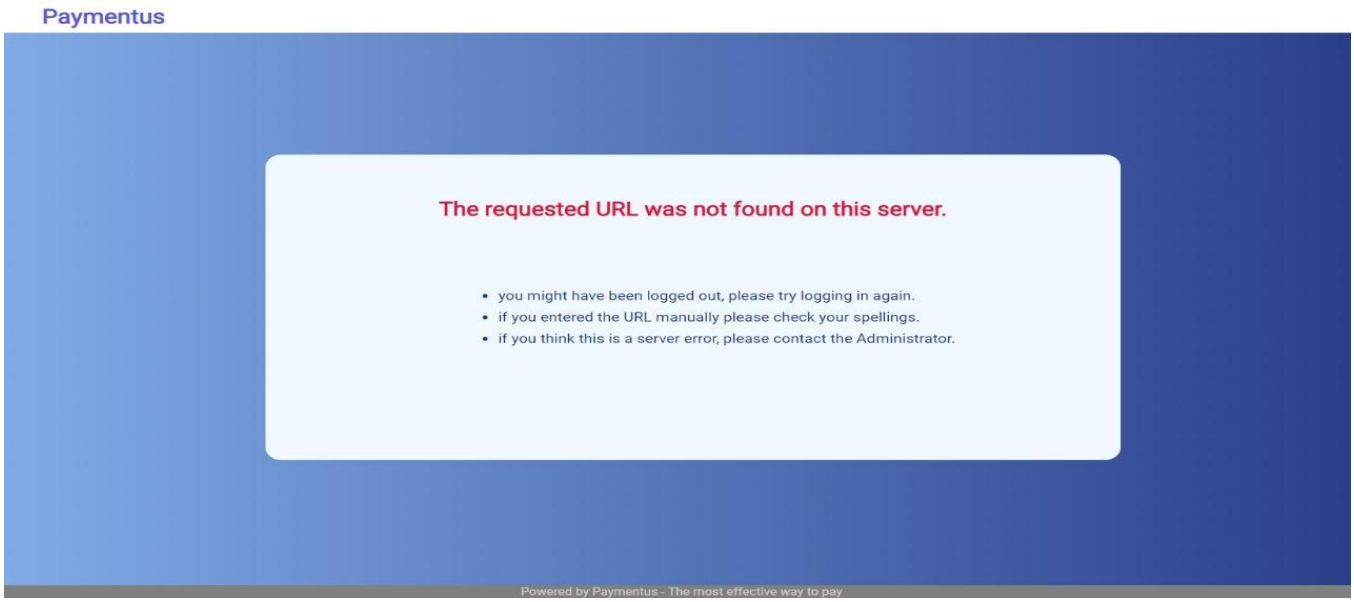


Fig 22 : Unauthorized access to the URL

If a user tries to access dashboard or a page which requires them to be logged in without logging in, they will see the above error page.

Conclusion

Perfect platform is a need of every user and Paymentus is trying to provide that quality experience and that best platform to all at genuine. Doing internship there made me realized that developer should not only care about the code but thinking about product is also important.

Product should be developed in such a manner that it should not impact the other parts of side. There should be a feeling of ease for user while using your product on platform. Scope of enhancement is always there, only thing required is developer seeing the product like user does.

Apart form all these code should be neat and we should always try to make parts which are reusable and help the code base in future

References

- https://paxcom.ai/
- https://www.paymentus.com/
- https://www.udemy.com/
- https://spring.io/
- https://reactjs.org/
- https://www.w3schools.com/REACT/DEFAULT.ASP
- https://www.baeldung.com/spring-tutorial
- https://www.baeldung.com/learn-jpa-hibernate
- https://www.javatpoint.com/java-jdbc