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: Shubham Sharma

181434

To

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 **Shubham 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 Shubham Sharma and Paxcom Pvt Ltd. Mohali, Punjab, dated 10 February, 2022.

Shubham Sharma
181434
Jaypee University of Information Technology
Waknaghat, Solan, H.P.

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

Dr. Kapil Sharma
Assistant Professor
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,

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 also like to thank **Dr. Kapil Sharma,** my project coordinator.

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. I would like to specifically mention my regards to Mr. Pankaj Chauhan (Sr. Software Engineer), Mr. Shikshank Sharma (Sr. Software Engineer) and Mr. Aashu Jaidka (Development manager) for their continuous support and guidance during this internship

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.

Shubham Sharma
181434
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.

Shubham Sharma

181434

Jaypee University of Information Technology

Waknaghat, Solan, H.P

INTERNSHIP REPORT UNDERTAKING

I, Shubham Sharma, Roll No. 181434, 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.

Shubham Sharma
181434
Jaypee University of Information Technology
Waknaghat, Solan, H.P

TABLE OF CONTENT

Content **Page No.**

PAXCOM INTRODUCTION

Paxcom:

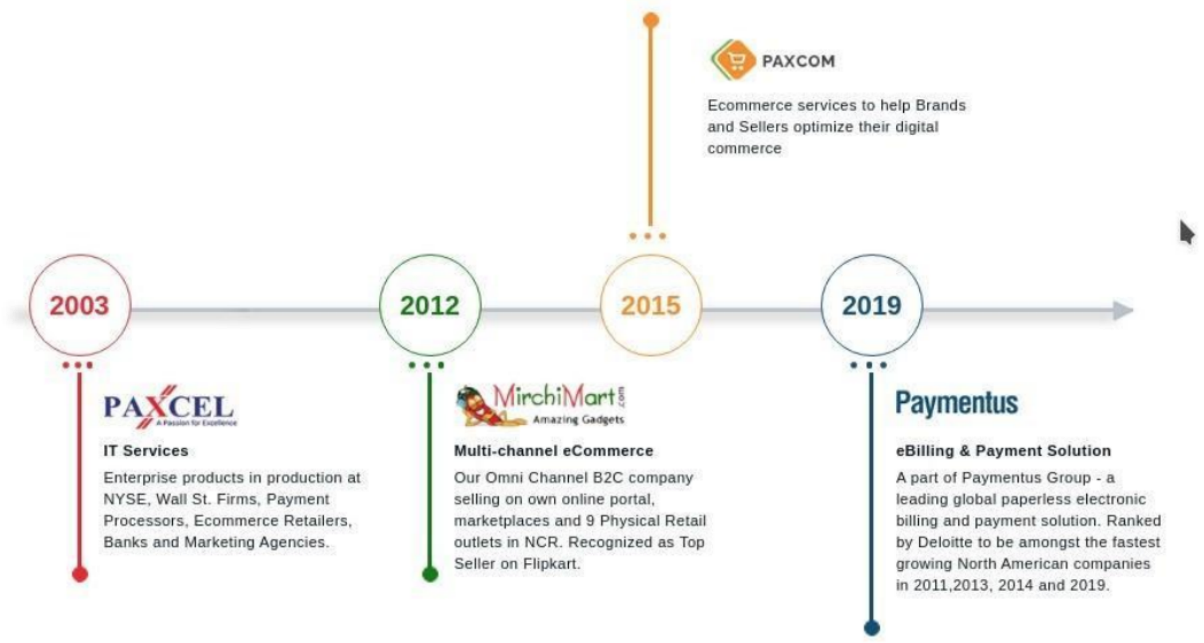
Paxcom is a SAAS-based platform that offers entire ecommerce automation solutions, including the creation and management of e-stores (or marketplaces), extensive data analysis, exponential revenue development, controlled warehouses and supply chains, and customer satisfaction. Paxcom is rapidly gaining ground. Every day, paxcom work with the world's most well-known businesses to help them strengthen their digital presence and increase sales. Paxcom's services and technologies may be customised to your needs, whether you need help troubleshooting or a comprehensive strategy. Everything paxcom does is driven by smart data analysis and analytics, allowing you to make educated decisions and fine-tune your product plans.

As of now Paxcom has offices in Washington, North Carolina, Delhi, Gurgaon and Mohali. Paxcom has a team of 400+ Ecommerce enthusiasts that are passionate about leveraging technology and our experience to make Digital Commerce easier for companies throughout the globe.

PepsiCo, Mondelez, Britannia, Lenovo, Wipro, and Abbott, among others, rely on our data, analytics, consulting, and solutions to guide and execute their ecommerce-driven retail strategy.

Paxcom is a subsidiary of Paymentus, a global leader in paperless electronic billing and payment solutions.

Paxcom Logo



PAXCOM

Ecommerce services to help Brands and Sellers optimize their digital commerce

PAXCEL
A Passion for Excellence

IT Services

Enterprise products in production at NYSE, Wall St. Firms, Payment Processors, Ecommerce Retailers, Banks and Marketing Agencies.

MirchiMart.com
Amazing Gadgets

Multi-channel eCommerce

Our Omni Channel B2C company selling on own online portal, marketplaces and 9 Physical Retail outlets in NCR. Recognized as Top Seller on Flipkart.

Paymentus

eBilling & Payment Solution

A part of Paymentus Group - a leading global paperless electronic billing and payment solution. Ranked by Deloitte to be amongst the fastest growing North American companies in 2011,2013, 2014 and 2019.

Paxcom's strength comes in having a strong blend of technology and ecommerce professionals working in tandem for a complete approach to its customers' growth, with over 400 people across 6 office locations. Paxcom is a subsidiary of Paymentus, a North American supplier of electronic billing and payment systems.

PAYMENTUS INTRODUCTION

Paymentus is the parent company of Paxcom pvt. Ltd. Paymentus was founded in 2004 as a result of a desire to change the way bills are paid. Paymentus has grown to become the top paperless electronic billing and payment system on the market, with over 1,700 clients, including some of the major billers in North America, because to its vision, innovation, and exceptional service.

We understand that in order to maintain our solutions current and relevant, we need employees that have the know-how, motivation, and aptitude for creating a fantastic client experience. Our dedicated, innovative staff transformed a concept into a secure, SAAS-based Customer Engagement and Payment Platform, allowing direct-bill companies to create a consistent customer experience while increasing adoption of cost-effective electronic billing and payment services.

Paymentus, which has been named one of Deloitte's fastest-growing North American firms for the past four years, is dedicated to developing better, quicker, more secure, and cost-effective billing and payment solutions. We are always looking for ways to provide more value to our clients, both in terms of solutions and service.

It's what's allowed us to develop so quickly in the previous ten years. When our clients prosper, we succeed as well. They prosper when their client ties are strengthened, and as a result, their consumers are more likely to use these cost-cutting technological services.

How Paymentus works

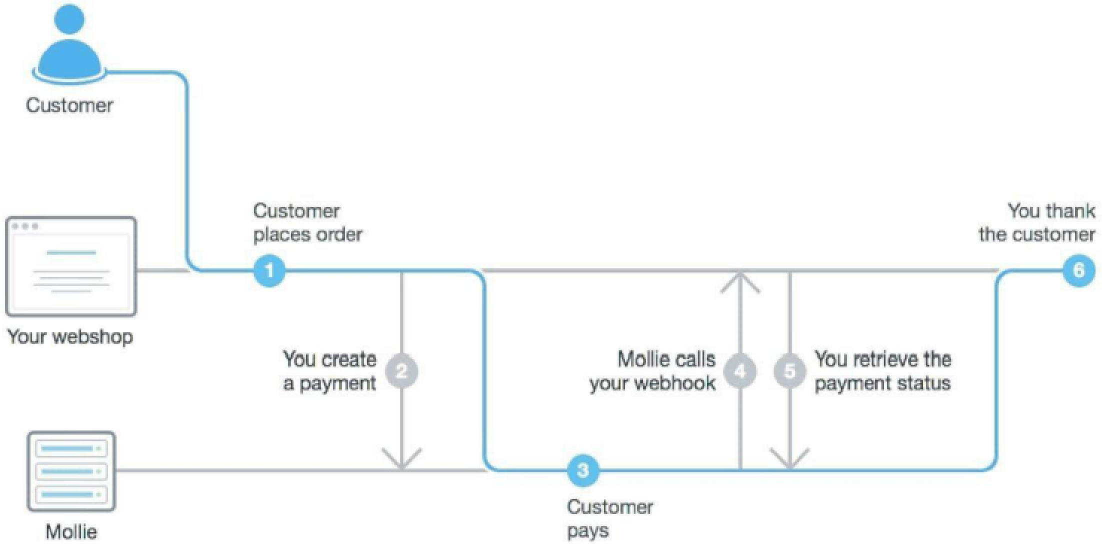
The very core functionality provided by paymentus is to simply make payments online hassle free. Paymentus products are widely used by companies to pay other companies.

How does it work?

To begin the payment procedure, type the URL of the organisation you wish to pay into your web browser and go to the billing area.

Are you unsure of the URL?

On the billing statement, it is frequently encountered. If not, numerous search engines such as Google and others can be used to find it.



About Products by Paymentus

Paymentus provides a variety of payment-related goods and services, including eBilling, invoice production, and easy payment gateway services.

The current team on which this internship is working is working on the paymentus dashboading product. The following products are more of a visualisation, analytics, and data querying code that allows the end user, be it a large business with a large number of clients and a high volume and velocity of transactions, or a single end common user, to basically access, analyse, and address financial questions from their account, or in some special test cases, also offer to answer certain business logics.

Agent Dashboard Systems:

This product gives our clients with specialised customer assistance via a dedicated dashboard that includes payment data, history visualisation, and other features. It also has functionality for IVR and Assisted IVR payments, as well as customer service.

Customer Portal:

This is a feature that serves as a payment option for clients by offering features such as wallets or a direct payment site for registered users.

ROTP (Remote One Time Payment):

This feature is essentially a payment gateway that allows guest users to make payments from a variety of sources and types. Works on the same principles as a payment gateway.

Description of the industry

Fintech (Financial technology):

FinTech is a combination of the words finance and technology, and it is essentially the amalgamation of the two sectors in which technology is utilised to develop and better novel domains in the financial world.

Everything is a subdomain of financial innovation aided by technology, from the ability for consumers or users to view and make financial transactions online, to applications that allow them to make financial transactions in seconds to anyone around the world, to means that enable financial institutions to make rapid financial transactions in large volume and velocity. FinTech's closest analogues in our daily lives include applications like Paytm, GPay, and others, as well as investing apps like Grow and Zerodha.

One of the most significant opportunities provided by the advent of fintech in the current marketspace is in terms of assisting the common man with financial literacy and financial accessibility; it has given people more and more control over their financial decisions and investments; and it has opened up investment opportunities that were previously out of reach of the common man.

The ability for smaller businesses to compete in the same arena as established banks and financial institutions has fueled FinTech's rise.

Thanks to FinTech, it's no longer about who's the biggest, but rather who's the fastest and most sensitive to changing customer expectations. Furthermore, fintech organisations' solutions are no longer "universal." Instead, it offers specialised (typically niche) services to meet specific financial demands. It might also be supplied at a far lower cost than typical financial services.

Electronic billing:

The practise of sending and paying bills electronically is known as electronic billing, or eBilling. Customers can get bills through email, online portal, or machine-readable data format, making distribution and payment more efficient. Accounts receivable manages the eBilling process by receiving findings from accounting software or an ERP system and delivering them to clients. Aspects of this paperless process, such as follow-up when invoice payment dates are delayed, are becoming increasingly automated to increase speed and accuracy.

With the localization of technology and an increase in the number of people using the internet and other forms of technology, more and more individuals are paying their bills online, resulting in hundreds of transactions every second and millions of invoices to be sent, cancelled, and kept.

In a variety of obvious ways, this industry transforms standard billing systems. It removes inefficiencies in processes, saves time and money, ensures more accurate and speedier invoice delivery to clients, and allows for rapid payments. A well-implemented electronic billing system is a strategic asset, not merely a cost-cutting measure.

On boarding

Like in any other company, on boarding at Paxcom includes various phases. After getting interviewed and receiving the offer letter, internship started on 7th February 2022, i.e the first day of work. On the very first day, an introductory session with the HR team takes places during which company's details and vision is told to the one's on boarding, HR welcomes them to the company. After that a meeting with the IT takes place which helps in setting up the office equipment like Laptops etc. IT guidelines needed to be followed by the employees are also told.

After that intern's manager contact the interns and welcomes them to the company. At early stages the intern is taught how to work and other technology stacks required by the team to carry out tasks.

Internship period was/is divided into various parts or phases, inclusive but not limited to Java Basics, Java Core with JDBC, Web Development basics, Spring Framework with spring MVC, Hibernate and JPA, MERN stack development.

Following part covers the tech stack learnt with the work done during this internship till date.

Java Basics

Java is a widely used computer language that was first released in 1995. Oracle owns Java, which is used on over 3 billion devices. Java is a platform as well as a programming language. Java is a high-level programming language that is also robust, object-oriented, and secure.

Java is an Object Oriented programming language which means that it provides the features of the same like, Encapsulation, Inheritance, Abstraction, Polymorphism. Java was widely used in programming the home appliances. Java is a platform independent programming language which means that you can write a java code on a windows machine and run it on linux machine.

Platform: A platform can be defined as hardware or software environment in which a particular programming language is going to work.

It is employed for the following purposes:

- Apps for mobile devices (specially Android apps)
- Applications for the desktop
- Applications for the web
- Web servers and application servers are two types of servers.
- Games
- Connection to a database

- Java may be used on a variety of systems (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most widely used programming languages on the planet. It is simple to learn and use.

Java Collections:

The collections framework is a set of useful interfaces and classes of data structures like Maps, Lists, and so on.

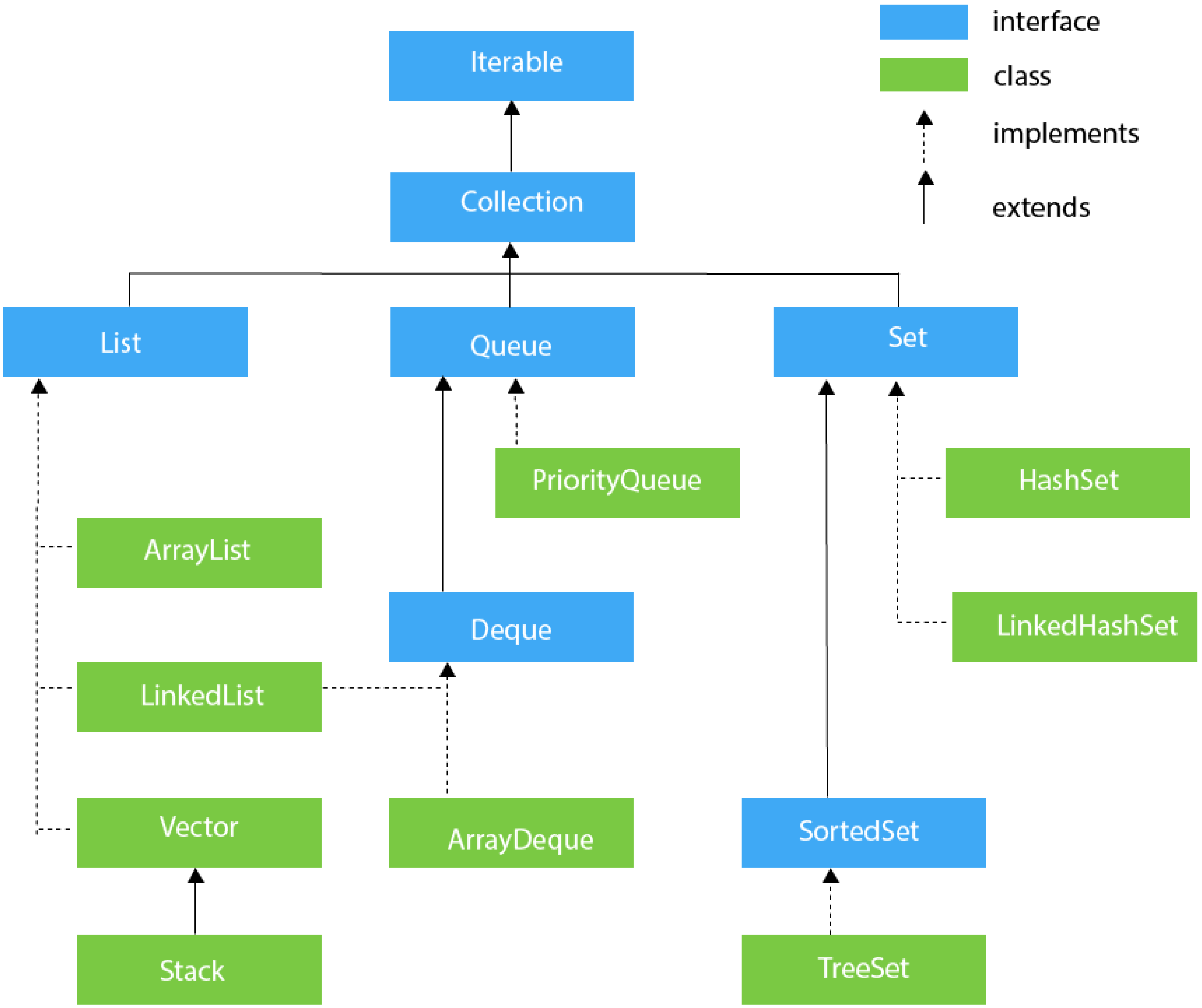
In Java, a collection is a framework that offers an architecture for storing and manipulating a collection of objects.

Many interfaces such as Set, List, Queue, Deque, and classes such as ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet, and others are available in the Java Collection framework.

Advantages of Collection Framework:

- No third-party API required
- No need to create API from handling collection objects
- Collection framework is tested and optimized

Hierarchy of collections framework:



Java Generics:

Generics is a generic programming feature that was added to Java in version J2SE 5.0 in 2004. It allows a type or function to work on objects of different kinds while maintaining compile-time type security.

Why are generics so popular?

All other classes are subclasses of Object, and an Object reference can refer to any object. There is no type safety in these features. Generics include this form of safeguard.

The generic allows us to specify a single declaration for all datatypes (not only primitives), but also for all Objects like Integers, Strings, and so on. As a result, there's no need to define the identical procedure for each data type, which is a waste of time and code. This also ensures that the code is safe at compilation time.

A generic type is a class or interface that takes types as parameters. The type parameter is specified using angle brackets (>).

Advantages

- Reusability (Once you've created a class or an interface, you may use it for any type.)
- Improved type safety and stronger type checking (reduces run time error)
- Type casting statements should be removed (this will increase the program's performance).

Java Core with JDBC

Java Database Communication, or JDBC, is a standard Java API allowing database-independent connectivity between the Java programming language and a variety of databases.

Each of the actions listed below that are often connected with database usage have APIs in the JDBC library.

- Making a database connection.
- Creating SQL or MySQL statements is a common task.
- In the database, running SQL or MySQL queries.
- Viewing and editing the records that result.

JDBC is a standard that defines a comprehensive set of interfaces for accessing an underlying database from a mobile device. Java may be used to create a variety of executables, including

- Applications written in Java
- Applets in Java
- Servlets in Java
- ServerPages in Java (JSPs)
- JavaBeans for Business (EJBs).

All of these executables can utilise a JDBC driver to connect to a database and utilise the information contained there.

JDBC is similar to ODBC in that it allows Java programmes to incorporate database-independent code.

Maven

Maven is a project management and comprehension tool that offers developers with a complete build lifecycle framework. The development team can quickly automate the project's build infrastructure thanks to Maven's standard directory design and default build lifecycle.

Maven can easily set up the way to work according to standards in the case of many development teams. Because most project setups are basic and reusable, Maven makes it easy for developers to set up reports, inspections, build, and testing automation settings.

Maven provides developers ways to manage the following −

- Builds
- Documentation
- Reporting
- Dependencies
- SCMs
- Releases
- Distribution
- Mailing list

To summary, Maven makes project development easier and more consistent. It smoothly manages duties including compilation, distribution, documentation, team communication, and more. Maven improves reusability by automating the majority of the build process.

Web Development

The term "web development" refers to the process of designing, developing, and managing websites. Web design, web publishing, web development, and database administration are all included. It is the building of an internet-based application, such as a website.

The term "web development" is made up of two words: "web" and "development."

Web: web pages, and anything else that functions through the internet are referred to as the web.

Development: is the process of creating an application from the ground up.

There are two methods to categorise web development:

- Frontend Development
- Backend Development

Frontend development: refers to the section of a website with which the user interacts directly. It's also referred to as the application's 'client side.'

Backend Development: The server side of a website is referred to as the backend. It is a section of the website that visitors are unable to see or interact with. It's the part of the software that doesn't interact with users directly. It is used to organise and store data.

Frontend Technologies:

HTML (HyperText Markup Language) is required for the creation of every website. This is a website's skeleton. No website can function without it. CSS (Cascading Style Sheets) beautifies the website, while JavaScript provides the brain that allows it to run. As a result, CSS and JavaScript bring beauty and intelligence to a website, respectively.

Let's look at another vehicle, this time an automobile. The HTML represents the car's metallic body, while the CSS represents the car's colour and design. Finally, a car's engine is similar to how a website's JavaScript adds functionality.

Backend Technologies:

A client or user typically sends a request to the webserver of the website he wishes to see. All of the files in the backend, which might be written in PHP, Python, or Node.js, are stored on the web server, which has its own IP address. The client receives HTML, CSS, and JavaScript as a response from the web server.

Frameworks for the back end: Back end frameworks include Express, Django, Rails, Laravel, Spring, and more.

Web Development using JAVA:

JSP:

JSP is a server-side programming language that allows developers to construct dynamic, platform-agnostic Web applications. JSPs may use the whole Java API set, including the JDBC API, to connect to corporate databases.

JavaServer Pages are commonly used in the same way that Common Gateway Interface-based applications are (CGI). JSP, on the other hand, offers several advantages over CGI.

- Performance is significantly improved since JSP allows Dynamic Elements to be integrated in HTML Pages rather than needing separate CGI scripts.
- Unlike CGI/Perl, which requires the server to load an interpreter and the target script each time the page is requested, JSPs are always generated before the server handles them.
- Because JSPs and Servlets are built on top of the Java Servlets API, they have access to all of the powerful Enterprise Java APIs like JDBC, JNDI, EJB, JAXP, and so on.

Servlets:

Servlets provide a component-based, platform-independent alternative for constructing Web-based applications without the performance limitations of CGI programmes. Servlets may use the whole Java API set, including the JDBC API, to connect to corporate databases.

Servlets can be used to accept user input via web page forms, display records from a database or another source, and dynamically build web pages.

Java Servlets are widely used for the same reasons as Common Gateway Interface programmes (CGI). Servlets, on the other hand, have several advantages over CGI.

- There has been a significant improvement in performance.
- Servlets execute in the address space of a Web server. It is not necessary to create a separate method for each client request.
- Servlets are platform independent since they are written in Java.
- The server's Java security manager enforces a set of limits to protect the server's resources. As a result, servlets are regarded as reliable.
- A servlet has access to all of the Java class libraries' capabilities. It connects to applets, databases, and other software using the sockets and RMI mechanisms you've already encountered.

Spring Framework

What exactly is Spring? - It is a Java framework for designing corporate applications that is lightweight, loosely connected, and integrated. - The Spring Framework is based on two design principles. Aspect Oriented Programming with Dependency InjectionIOC, AOP, DAO, Context, ORM, WEB MVC, and other modules make up the Spring framework.

The desire to perform things in a simple, secure, and effective manner is more than enough reason to utilise spring. - It's been discovered that there's a lot of code complexity in development that may be eliminated, such as transaction handling, JDBC, ORM interaction, Exception Handling, Testing, and so on. - Springs provides compatibility for existing J2EE technologies and makes development simple and efficient.

Spring framework has a number of advantages, including: - It is lightweight and has a low overhead when used for development. - Use Dependency Injection or Inversion of Control to create components that are independent of one another, with Spring Container handling the wiring to complete our task. - The Spring IoC container is in charge of managing the Spring Bean life cycle as well as project-specific parameters like JNDI lookup. - The Spring MVC framework may be used to build web apps and restful web services that can deliver XML and JSON responses. - Support for transaction management, JDBC operations, file uploading, exception handling, and other features with minimal configuration, using annotations or the spring bean configuration file.

The following are some of the benefits of adopting Spring Framework: - Reducing direct dependencies between application components; the Spring IoC container is generally in charge of initialising resources or beans and injecting them as dependencies. - Spring framework makes writing unit test cases simple because our business logic has no direct dependency on resource implementation classes. For testing reasons, we can easily construct a test configuration and inject our fake beans.

Dependency Injection (DI)

The Dependency Injection (DI) flavour of Inversion of Control is the technique with which Spring is most closely associated. The idea of inversion of control (IoC) is a broad one that may be stated in a variety of ways. Inversion of Control is manifested in a variety of ways, one of which being Dependency Injection.

When building a complicated Java programme, application classes should be as independent as feasible from other Java classes so that they may be reused and tested separately from other classes during unit testing. Dependency Injection aids in the bonding of these classes while maintaining their independence.

What precisely is dependency injection? Let's take a closer look at these two terms independently. The dependent element is translated into a relationship between two classes in this case. Class A, for example, is reliant on class B. Let's have a look at the second section, injection. This simply indicates that the IoC will inject class B into class A.

Dependency injection can take place during construction by giving arguments to the function Object() { [native code] } or afterward by utilising setter methods.

Aspect Oriented Programming (AOP):

Aspect Oriented Programming (AOP) is a novel approach to creating real-time applications that include all necessary services.

The AOP module will allow us to separate services from business logic as well as inject services for business logic at runtime.

The services are also known as? Cross-Cutting Concerns? in spring language (Middle ware services).

Through its configuration management capability, the Spring AOP module incorporates aspect-oriented programming features directly into the Spring framework. As a result, every object maintained by the Spring framework may be simply AOP-enabled. Transaction management services for objects in any Spring-based application are provided by the Spring AOP module. You may use Spring AOP to add declarative transaction management to your applications without having to rely on EJB components.

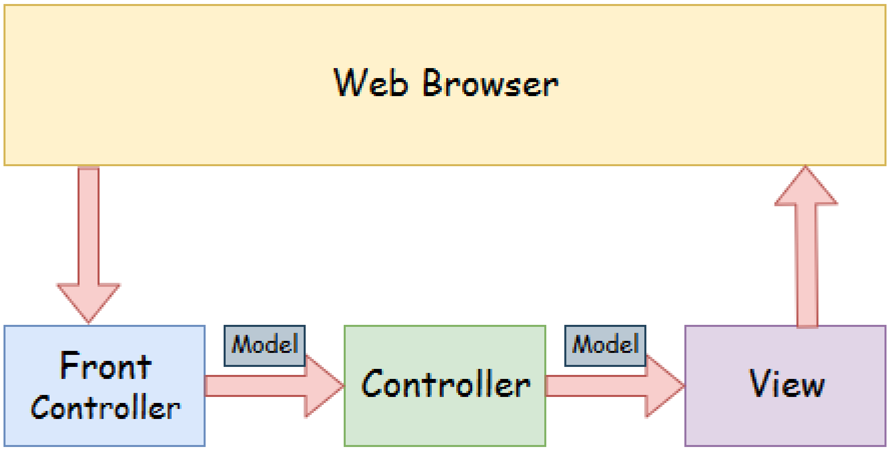
Spring – MVC

It's a Spring framework module that deals with the Model-View-Controller (MVC) paradigm, as the name implies. It combines all of the benefits of the MVC paradigm with Spring's ease of use.

Spring uses its DispatcherServlet to build MVC using the front controller design.

In a nutshell, the DispatcherServlet serves as the primary controller for routing requests to their proper destinations. The view is represented by any of the numerous template engines, and the model is nothing more than our application's data.

Spring Web Model-View-Controller:

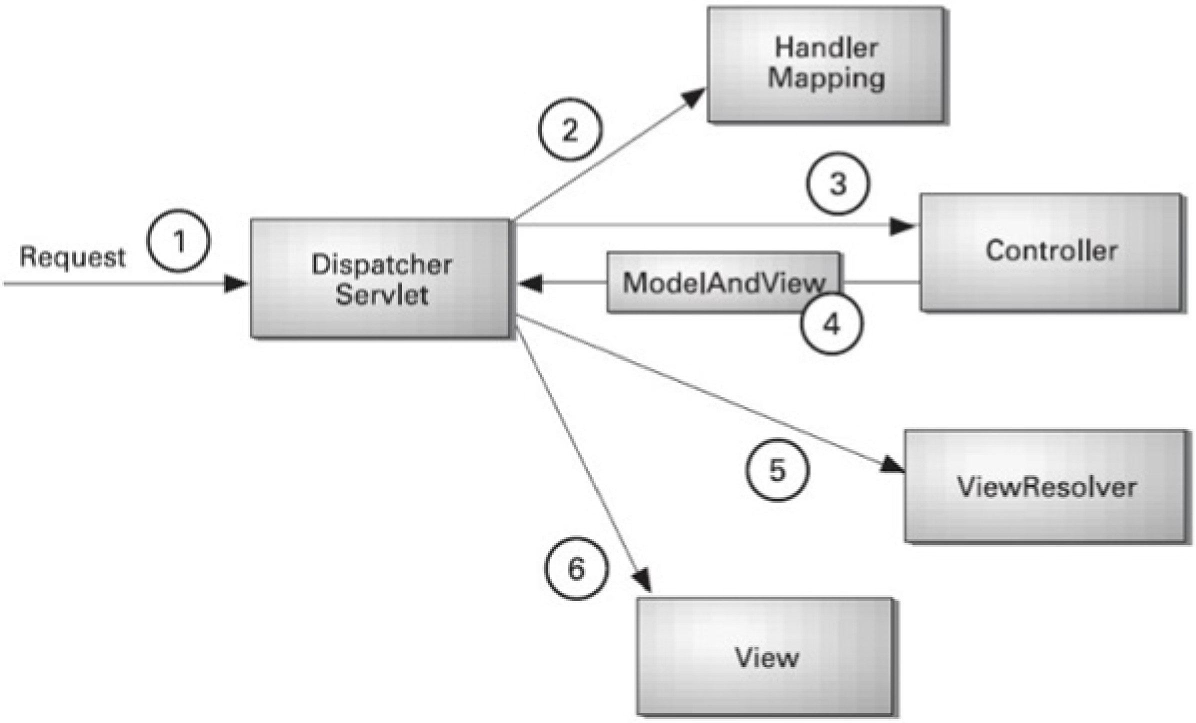


Model-view-controller (MVC) is a well-known design paradigm for creating user interfaces. By separating the responsibilities of model, view, and controller in an application, it primarily decouples business logic from UIs. Models are often in charge of encapsulating application data for views to convey. Views should just display this data and should not include any business logic. Controllers are in charge of handling user requests and activating back-end services (manager or dao) to handle business logic.

Back-end services may return some data for views to display after processing.

Controllers gather this information and build models for presentation. The MVC pattern's main principle is to segregate business logic from user interfaces so that they may change independently without impacting one other.

Flow of Spring Web MVC:



- As shown in the diagram, the DispatcherServlet, which serves as the front controller, intercepts all incoming requests.
- The DispatcherServlet reads the XML file for a handler mapping entry and sends the request to the controller.
- The controller returns a ModelAndView object.
- The DispatcherServlet looks in the XML file for a view resolver item and then calls the given view component.

Hibernate

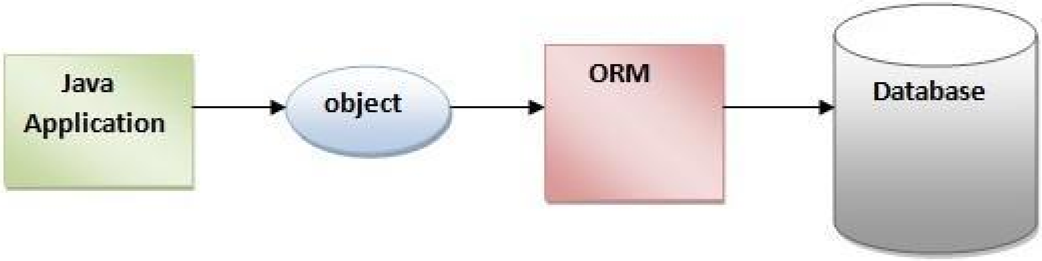
In the Java world, Hibernate is one of the most prominent Object/Relational Mapping (ORM) frameworks. It allows programmers to link the object structures of standard Java classes to a database's relational structure. The job of storing data from object instances in memory to a persistent data store and loading it back into the same object hierarchy becomes substantially easier with the aid of an ORM framework.

Simultaneously, ORM solutions like as Hibernate seek to abstract from the specific data storage product. This eliminates the need to build code that handles the tiny changes between the supported database products while utilising the same Java code.

Hibernate is also a Java Persistence API (JPA) provider, which implies it implements the API (JPA). JPA is a vendor-agnostic standard for mapping Java objects to relational database tables. Because another article in the Ultimate series already covers JPA, this one concentrates on Hibernate and so avoids using JPA annotations in favour of Hibernate-specific configuration files.

What Is An ORM Tool, Exactly?

It's a strategy for mapping the objects in a database. Data generation, modification, and access are all made easier using an ORM tool. It interacts with databases via the Java API internally.



A relational database is represented by rows and columns in a tabular format, but Java is an object-oriented programming language that is represented by a network of objects. So, there's an apparent mismatch when it comes to storing an object in a database, and we have Object Relational Mapping to handle it (ORM). ORM is a framework for mapping an object to a relational database as a result.

Hibernate overcomes the drawbacks of previous technologies such as JDBC. Let's look at how it outperforms JDBC in terms of job optimization.

- Hibernate eliminates the database requirement that JDBC has.
- Working with JDBC, changing databases costs a lot of money; hibernate solves this problem with ease.
- When working with JDBC, which is readily handled by hibernate, code portability is not an option.
- The object-level connection is strengthened via Hibernate.
- It gets rid of the exception-handling portion, which is required when dealing with JDBC.
- The object level connection is overcome by Hibernate.
- By addressing the boiler plate problem, it minimises the amount of code while increasing readability.

Hibernate overcomes all of JDBC's flaws to give optimum and efficient solutions for each activity.

Because it is an open-source framework, anyone can use it for free. The source code for hibernation may be available on the internet, allowing for changes.

The benefit of having a lightweight framework can be seen in the significantly reduced installation package. When no container is used for execution, efficiency improves. Even while hibernation can function with numerous technologies at the same time, that doesn't imply it couldn't work on its own. We may also operate with hibernation without using any other technology.

Hibernate is unique in that it doesn't need to implement hibernate API interfaces or extend from hibernate API classes since hibernate application development classes are loosely connected.

Functionalities Supported By Hibernate:

- Hibernate is database agnostic thanks to the Hibernate Query Language.
- It facilitates DDL operations that are performed automatically.
- Auto Primary Key Generation is supported by Hibernate.
- Cache memory is supported.
- Hibernate does not need exception handling.
- Hibernate is an ORM tool, which is the most significant feature.

The Advantages Of Hibernate In Java

• Lightweight and open-source — Its low weight and open-source nature help with accessibility and efficiency.

• Improved performance – Making use of cache memory boosts performance.

• Database Independence — Because of its database independence, it can work with a wide range of databases.

• Using auto DDL procedures, we can construct tables without having to manually create them.

• It uses XML files to bind Java classes to databases, eliminating the need to write any code.

• We can store and get data directly from the database using basic APIs.

• The use of an application server is not required.

• Database access is reduced using smart fetching techniques.

• It enables simple data queries.

JPA- Java Persistence API

JPA (Java Persistence API) is a Java specification. It's used to keep data from a Java object and a relational database in sync. JPA serves as a connector between object-oriented domain models and relational database management systems.

JPA is merely a standard, thus it doesn't accomplish anything on its own. It needs to be put into action. As a result, ORM technologies like as Hibernate, TopLink, and iBatis support JPA data persistence requirements.

Database operations are performed by every corporate programme by storing and retrieving large volumes of data. Despite all of the storage management solutions available, application developers frequently struggle to perform database activities effectively.

To communicate with the database, Java developers often require a lot of code or a proprietary framework, however with JPA, the effort of dealing with the database is substantially reduced. It connects object models (Java programmes) with relational models (database program).

What exactly is JPA?

The Java Persistence API is a set of classes and methods for persistently storing large volumes of data in an Oracle database.

A programmer uses the 'JPA Provider' framework, which permits easy interaction with database instances, to alleviate the load of developing scripts for relational object management. JPA takes over the needed framework in this case.

Project Work

Requirements:

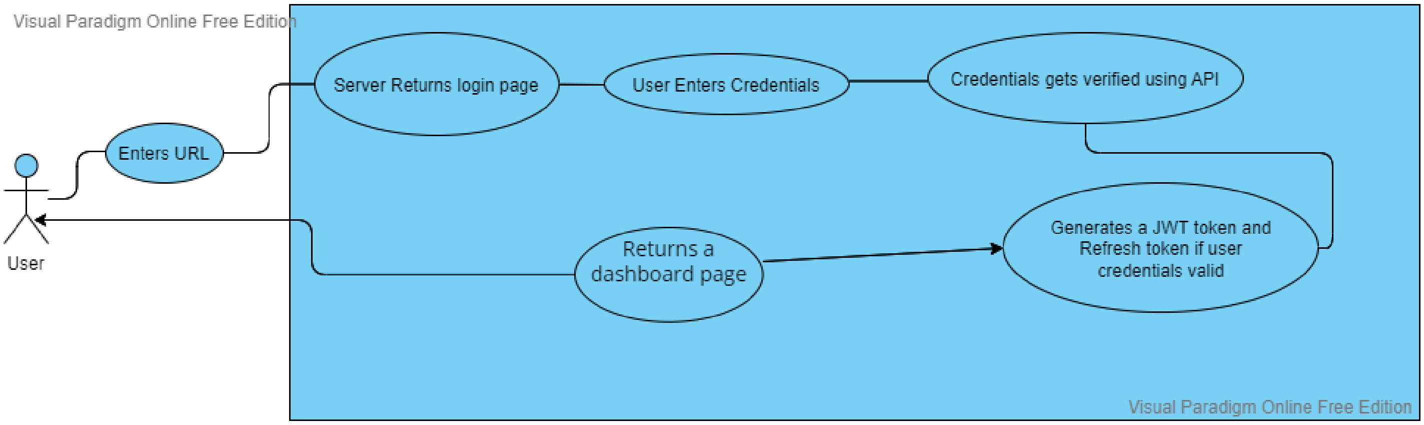
Software Requirements:

- HTML, CSS
- JavaScript (Fundamental and Core for React.js)
- React.Js
- Node.js
- MongoDB
- VS Code
- Web Browser

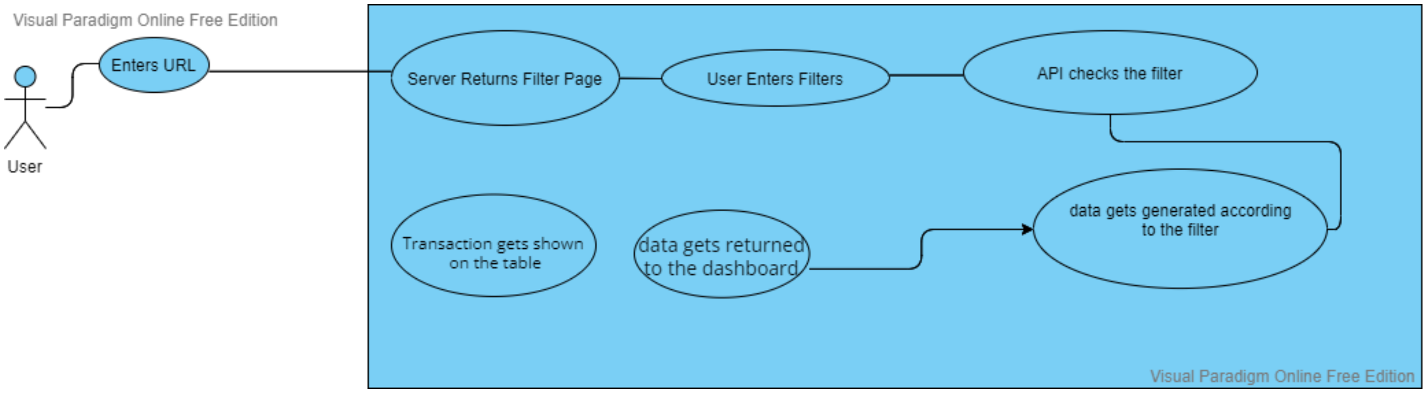
Hardware Requirements:

- 2.2Ghz CPU or higher
- 8GB RAM or higher
- 20GB HDD/SSD

Diagrams:


Visual Paradigm Online Free Edition
Enters URL
User
Server Returns login page
User Enters Credentials
Credentials gets verified using API
Generates a JWT token and Refresh token if user credentials valid
Returns a dashboard page
Visual Paradigm Online Free Edition

Login DFD


Visual Paradigm Online Free Edition
Enters URL
User
Server Returns Filter Page
User Enters Filters
API checks the filter
data gets generated according to the filter
data gets returned to the dashboard
Transaction gets shown on the table
Visual Paradigm Online Free Edition

Data filter DFD

JavaScript Essentials for React.js:

Arrow Functions and Function Declarations:

The component is the foundation of every React application. Components in React are defined using JavaScript functions and classes.

React components, unlike JavaScript functions, return JSX elements, which are utilised to organise our application's interface. Camel case is used for JavaScript functions, whereas pascal casing is used for React function components (in which all words are capitalized). In React, function components may be written using both function declarations and arrow functions.

The main advantage of arrow functions is their conciseness. We may utilise many shorthands to create our functions in order to eliminate unneeded boilerplate, and we can even do it on a single line.

Template Literals:

Working with strings in JavaScript has a sloppy past, especially when you wish to concatenate or join numerous strings. To join strings before the advent of ES6, you had to use the + operator to join each section of the string together.

With ES6, we now have a new type of string called a template literal, which consists of two back ticks " instead of single or double quotes.

The flexibility to utilise any JavaScript expression (that is, anything in JavaScript that resolves to a value) within the $ syntax is what makes template literals so powerful.

Short Conditionals: &&, ||, Ternary Operator:

Given that React is merely JavaScript, selectively displaying (or hiding) JSX components is as simple as using basic if and switch expressions. We may reduce redundancy and make our code more succinct by using several fundamental JavaScript operators.

Using the ternary operator, we can change the if expression into the following. The ternary operator is similar to an if-statement in that it is shorter, acts as an expression rather than a statement, and can be used within JSX.

Three Array Methods: .map(), .filter(), .reduce():

Using curly brackets to insert primitive values into JSX components is simple.

Any valid expressions can be inserted, including variables with primitive values (strings, integers, booleans, and so on) and object attributes with primitive values. What if we have an array and want to loop through it to display each array element within a single JSX element?

The.map() function may be used for this. It allows us to alter each element in our array according to the inner function's instructions.

It's worth noting that when combined with an arrow function, it's very succinct.

As the name implies,.filter() allows us to remove certain members from our array. For example, we might use.filter to delete all names of programmers that began with the letter "J." ().

It's vital to remember that.map() and.filter() are merely various forms of the.reduce() array function, which may convert array values into almost any data type, including non-array values.

Object Tricks: Property Shorthand, Destructuring, Spread Operator:

Objects, like arrays, are a data structure that you must be familiar with when using React.

Unlike arrays, objects exist for the purpose of organising key-value storage, therefore you'll need to be extremely familiar with accessing and modifying object attributes.

You name the property and its related value when you add properties to an object as you construct it. If the property name and the value are the same, all you have to do is list the property name.

The dot notation is the most common technique to obtain properties from an object. Object destructuring, on the other hand, is a more practical method. It allows us to extract attributes from a given object into independent variables with the same name.

It seems as though you're writing an item backwards, which is part of what makes the process natural. It's a lot more convenient than having to type the object name numerous times each time you want to get a value out of it.

Promises + Async/Await :

Asynchronous code — code that takes an indeterminate amount of time to run – is found in almost every React application. Especially if you're utilising browser capabilities like the Fetch API or the third-party package axios to access or alter data from an external API.

Promises are used to resolve asynchronous code so that it may be read from top to bottom like conventional synchronous code.

Callbacks have typically been used to resolve asynchronous code in Promises. The.then() callback is used to handle promises that have been successfully resolved, whereas the.catch() callback is used to resolve promises that have returned an error.

Instead of constantly needing to utilise callbacks to resolve our data from a promise, we can use the async/await syntax, which has been tidied up to seem like synchronous code.

Only functions can utilise the async and await keywords (normal JavaScript functions, not React function components).

ES Modules + Import / Export syntax:

We were able to easily exchange code across our own JavaScript files and third-party libraries utilising ES modules thanks to ES6.

We can also use assets like photos and svgs, as well as CSS files, as dynamic values in our code when we use tools like Webpack.

The purpose of ES modules is to be able to break up our JavaScript code into multiple files so that it may be reused across our programme.

Variables and functions can be imported and exported in JavaScript programming. Importing and exporting may be done in two ways: named imports/exports and default imports/exports.

React.js:

ReactJS is an open-source, component-based front-end library that is only responsible for the application's display layer. Facebook is in charge of keeping it up to date.

To fill in data (views) in the HTML DOM, ReactJS employs a virtual DOM-based technique. Because it just modifies individual DOM components rather than rebuilding the whole DOM, the virtual DOM works quickly.

A React application consists of several components, each of which is responsible for producing a tiny, reusable piece of HTML. Complex applications may be developed using simple building blocks by nesting components within other components. A component can also keep track of its own internal state; for example, a TabList component can keep track of the currently open tab by storing a variable.

We may create components in React using JSX, a domain-specific language. We can use HTML to construct our components and mix in JavaScript events using JSX. This will be converted to a virtual DOM by React, which will then output our HTML for us.

By employing the virtual DOM, React "reacts" to state changes in your components fast and automatically to rerender the components in the HTML DOM. The virtual DOM is a representation of an actual DOM that is stored in memory. React can act rapidly by executing most of the work within the virtual DOM rather than directly in the browser's DOM, and only adding, updating, and removing components that have changed since the last render cycle.

JSX, a domain-specific language, is used to build components in React. We can use HTML to construct our components and mix in JavaScript events with JSX. This will be converted to a virtual DOM by React, which will then output our HTML.

By employing the virtual DOM, React "reacts" to state changes in your components swiftly and automatically, rerendering them in the HTML DOM. A virtual DOM is a representation of a real DOM that is stored in memory. React can act rapidly by executing the majority of the work within the virtual DOM rather than the browser's DOM, and only adding, updating, and removing components that have changed since the last render cycle.

Project introduction: 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.

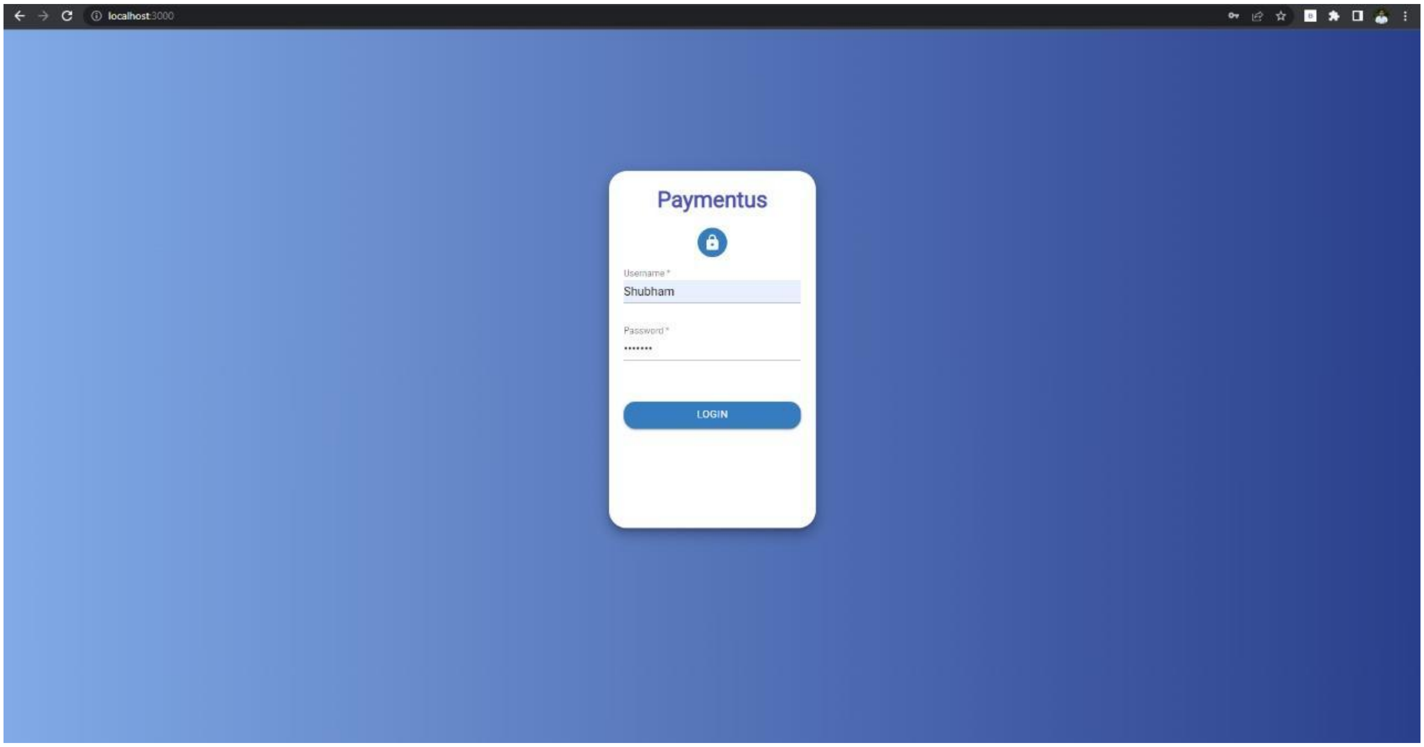
Step-by-step Working:

- 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.

Login Functionality:



Username *
Shubham

Password *
•••••••

Fig 1: Login page

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:

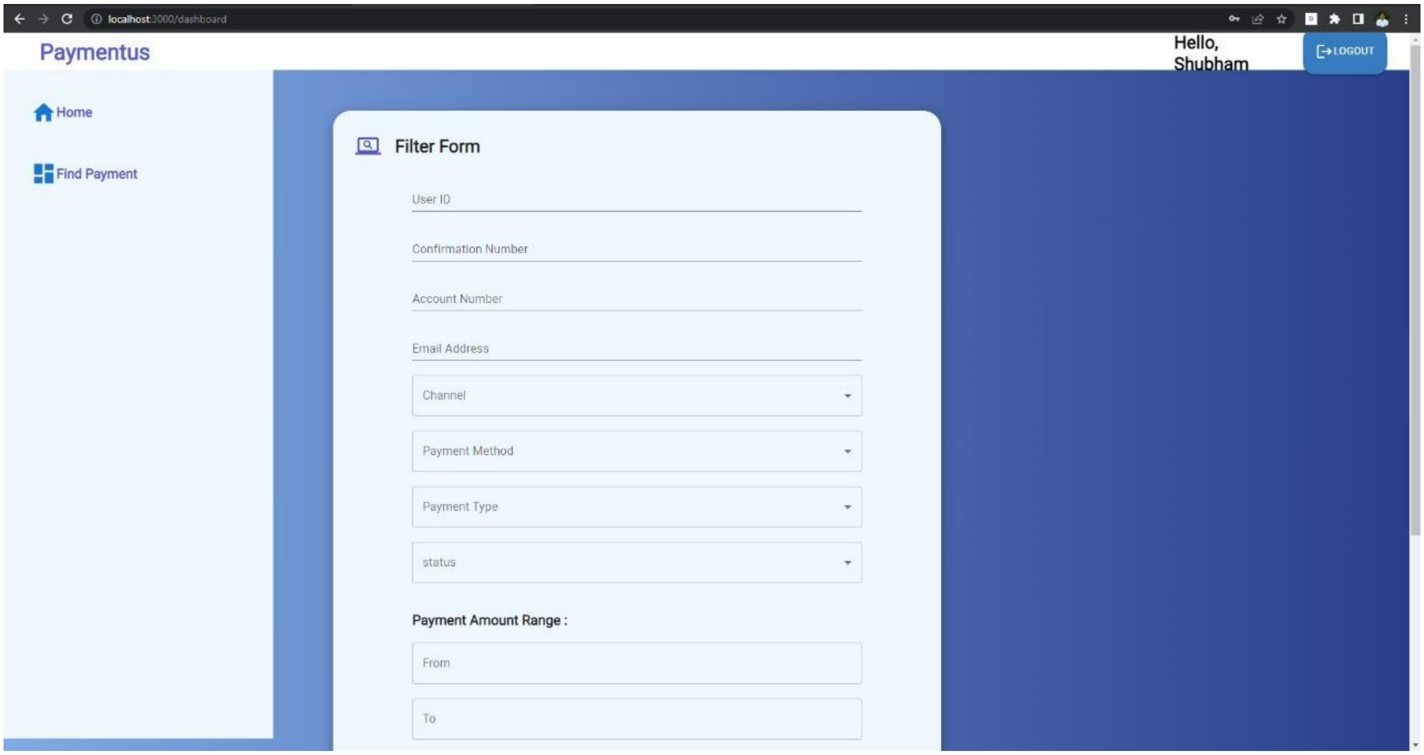
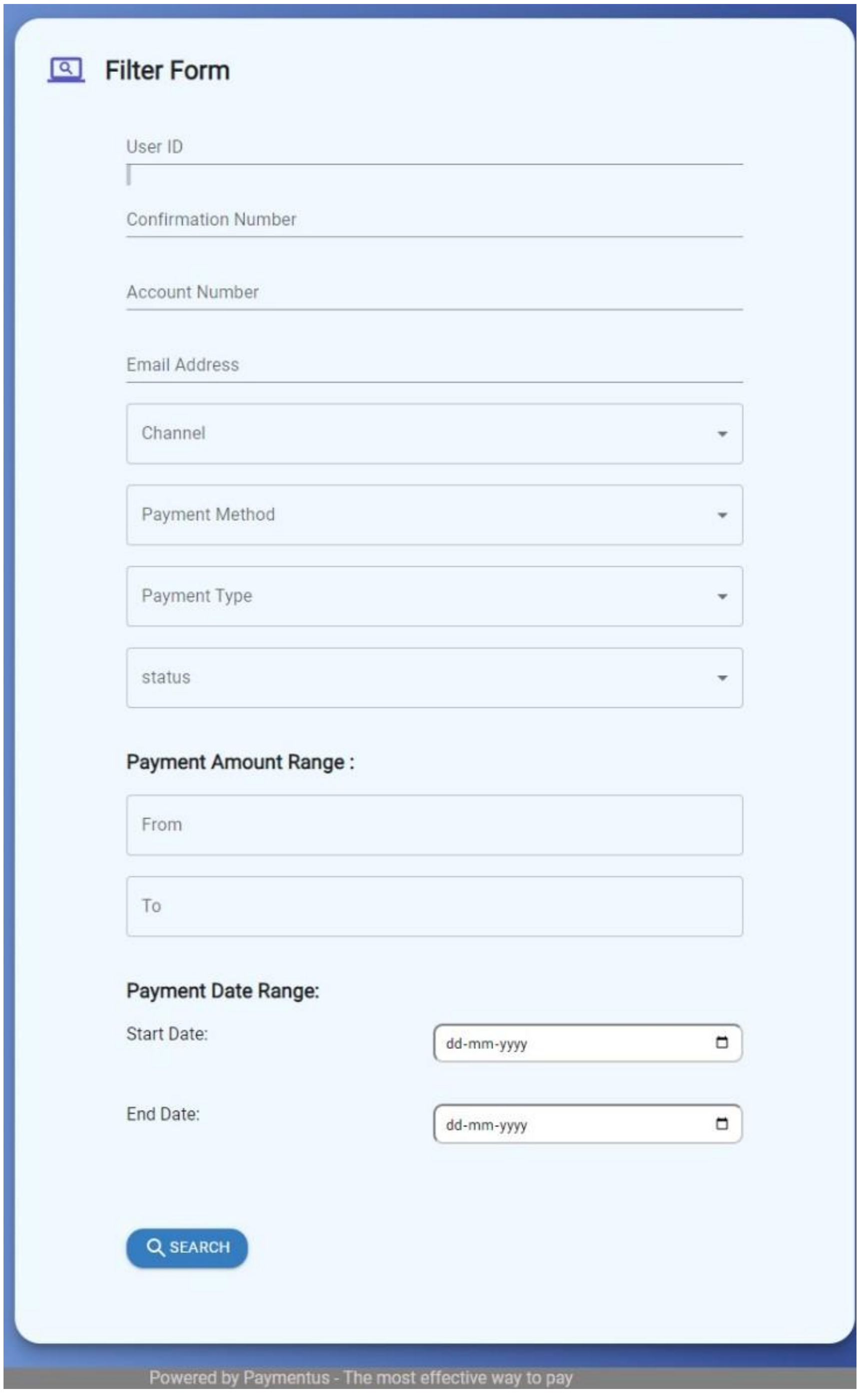


Fig 2: Dashboard

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:



Filter Form

User ID

Confirmation Number

Account Number

Email Address

Channel ▾

Payment Method ▾

Payment Type ▾

status ▾

Payment Amount Range :

From

To

Payment Date Range:

Start Date: dd-mm-yyyy

End Date: dd-mm-yyyy

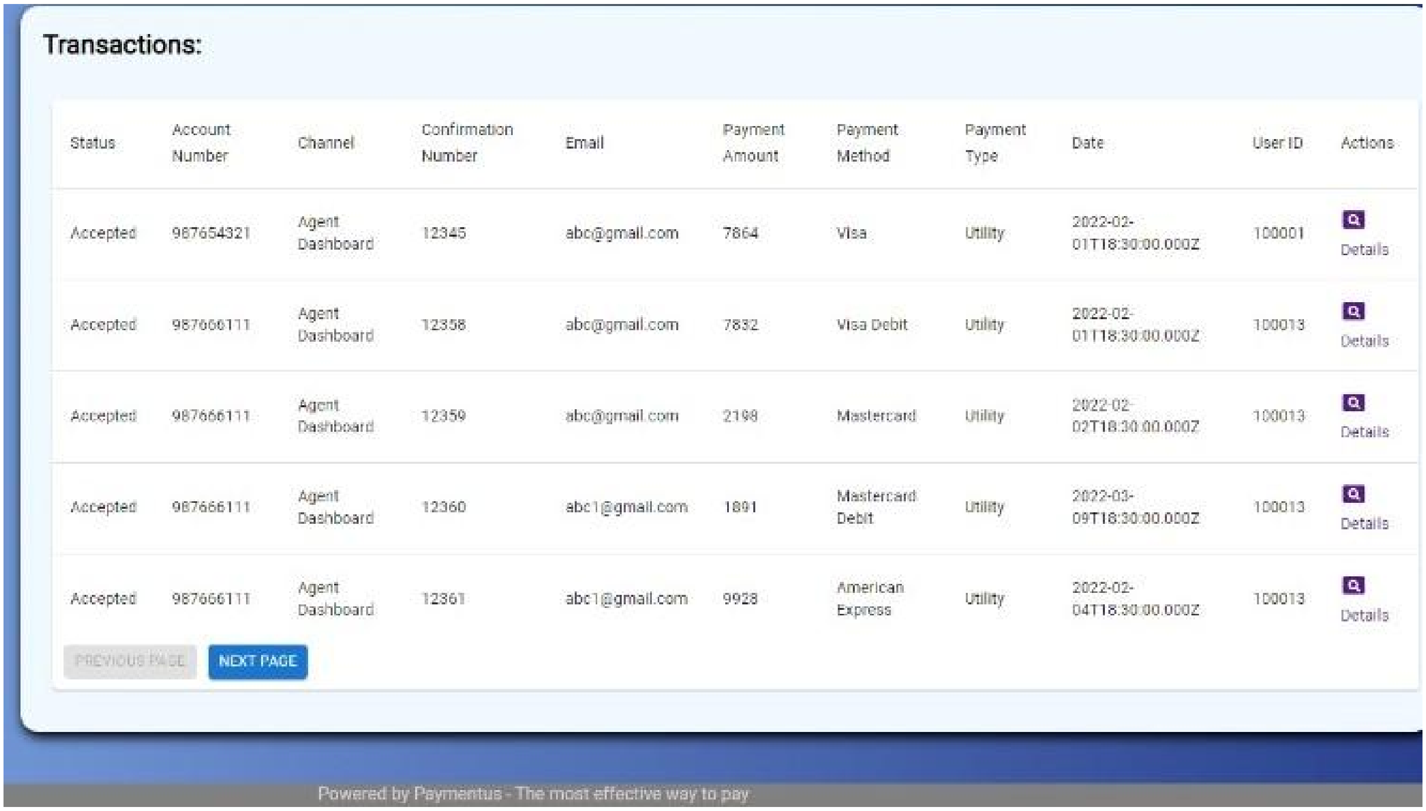
SEARCH

Powered by Paymentus - The most effective way to pay

Fig 3: Filter form

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

Transaction table functionality:



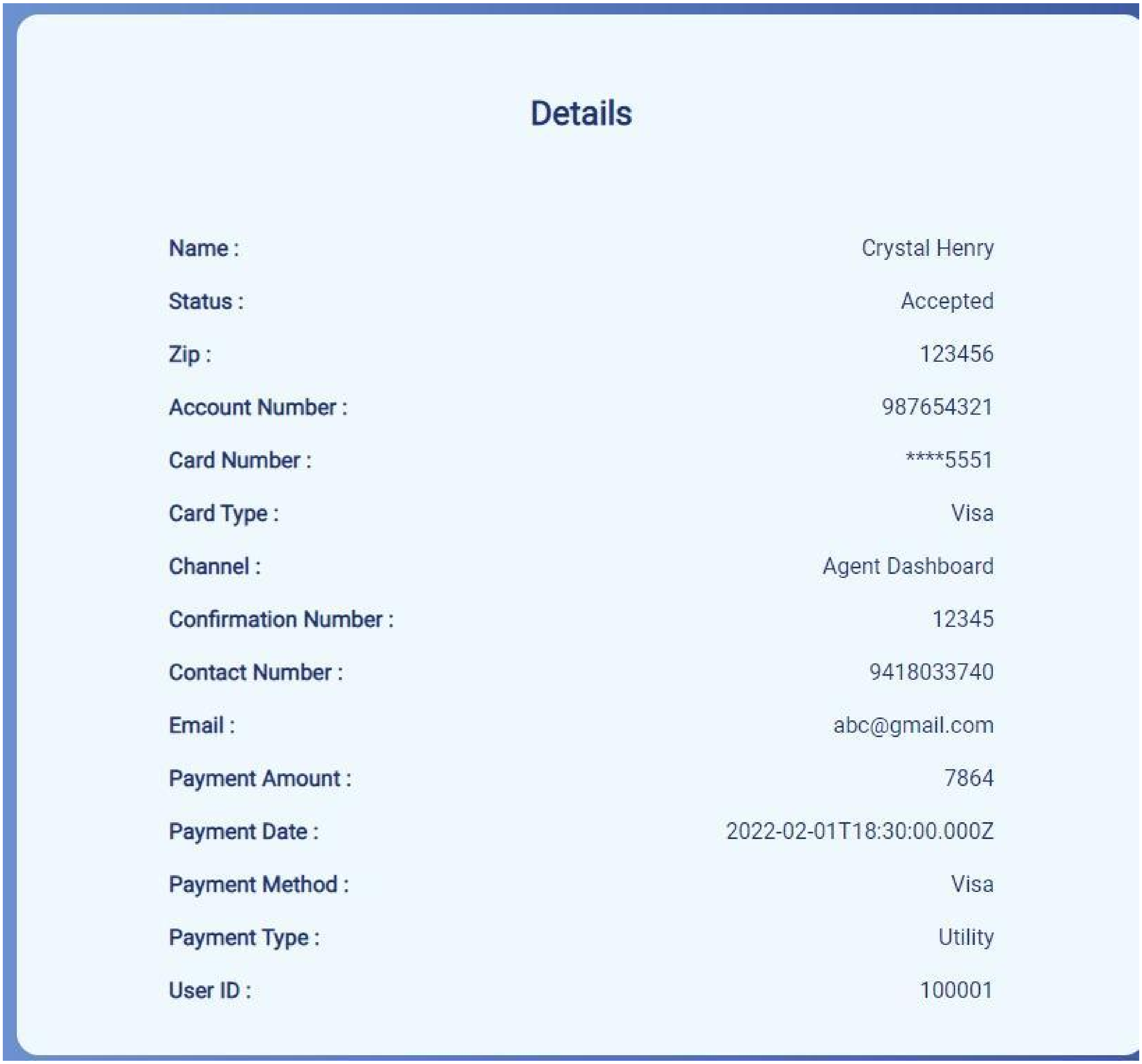
Transactions:

Status	Account Number	Channel	Confirmation Number	Email	Payment Amount	Payment Method	Payment Type	Date	User ID	Actions
Accepted	987654321	Agent Dashboard	12345	abc@gmail.com	7864	Visa	Utility	2022-02-01T18:30:00.000Z	100001	Details
Accepted	987666111	Agent Dashboard	12358	abc@gmail.com	7832	Visa Debit	Utility	2022-02-01T18:30:00.000Z	100013	Details
Accepted	987666111	Agent Dashboard	12359	abc@gmail.com	2198	Mastercard	Utility	2022-02-02T18:30:00.000Z	100013	Details
Accepted	987666111	Agent Dashboard	12360	abc1@gmail.com	1891	Mastercard Debit	Utility	2022-03-09T18:30:00.000Z	100013	Details
Accepted	987666111	Agent Dashboard	12361	abc1@gmail.com	9928	American Express	Utility	2022-02-04T18:30:00.000Z	100013	Details

Fig 4: 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:



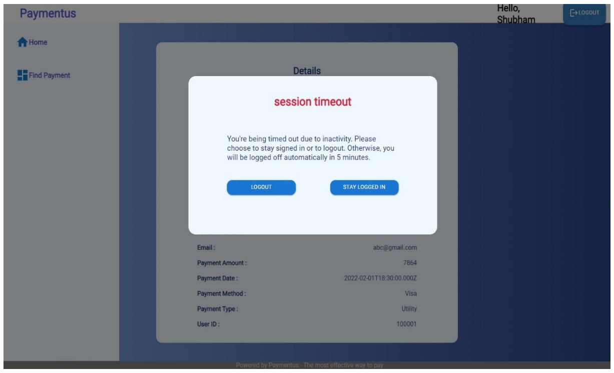
Details

Name :	Crystal Henry
Status :	Accepted
Zip :	123456
Account Number :	987654321
Card Number :	****5551
Card Type :	Visa
Channel :	Agent Dashboard
Confirmation Number :	12345
Contact Number :	9418033740
Email :	abc@gmail.com
Payment Amount :	7864
Payment Date :	2022-02-01T18:30:00.000Z
Payment Method :	Visa
Payment Type :	Utility
User ID :	100001

Fig 5: Transaction details

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:



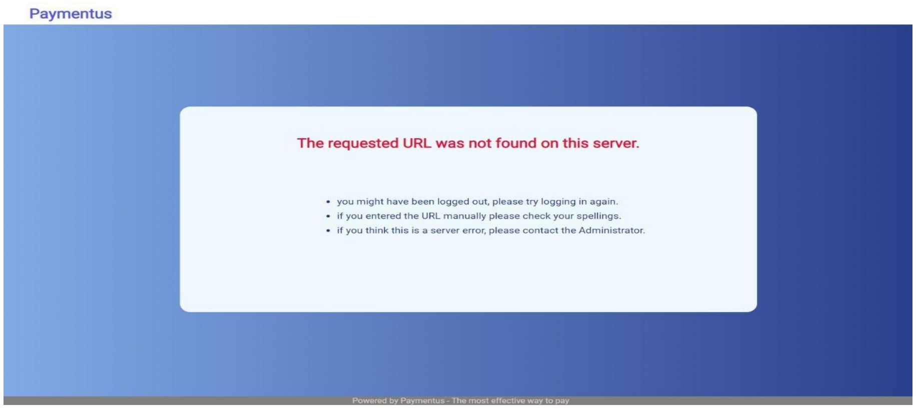
session timeout

You're being timed out due to inactivity. Please choose to stay signed in or to logout. Otherwise, you will be logged off automatically in 5 minutes.

Email : abc@gmail.com
Payment Amount : 7864
Payment Date : 2022-02-01T18:30:00.000Z
Payment Method : Visa
Payment Type : Utility
User ID : 100001

Fig 6: Session timeout

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



The requested URL was not found on this server.

- you might have been logged out, please try logging in again.
- if you entered the URL manually please check your spellings.
- if you think this is a server error, please contact the Administrator.

Fig 7: Unauthorized access to the URL

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

Results

This internship was a wealth of information; not only did I gain knowledge in Full Stack Development, but I also learned about how a project is developed, how a team works, how each employee's work is tracked, how work is distributed among different team members, what are the various stages of development, what are the technical problems that one encounters during the development of a project, and what all is required before the development begins. During the internship I’m getting the opportunity to worth with the finest software engineers in the industry.

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