

“ DIGITAL WALLET ”

A project report submitted

In partial fulfillment the award of the degree of

BACHELOR OF TECHNOLOGY

Submitted

By

D.A. GAYATRI	(19B61A0513)
D.BHANU PRATAP	(19B61A0512)
M.JYOTSNA HARIKA	(19B61A0530)
B.V.S POOJITHA	(19B61A0505)
G. DURGA NAIDU	(19B61A0519)

Under The Esteemed Guidance Of

Mr.T.ANJI KUMAR ,Mtech , (Ph.D), Assoc. Professor



SATYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

GAJULAREGA, VIZIANAGARAM, Andhra Pradesh-535002.

Accredited by “NAAC”

Approved by AICTE & Affiliated to JNTU, KAKINADA

(2019-2023)

SATYA INSTITUTE OF TECHNOLOGY & MANAGEMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Project entitled “**DIGITAL WALLET**” is a bonafide record of work done by **D.A.GAYATRI (19B61A0513), D.BHANU PRATAP (19B61A0512), M.JYOTSNA HARIKA (19B61A0530), B.V.S POOJITHA (19B61A0505), G.DURGA NAIDU (19B61A0519)** Department of CSE, Satya Institute of Technology and Management, during the year 2019-2023 in partial fulfillment of the requirements for the award of Bachelor of Technology by JNTUK.

Internal Guide

Mr.T.ANJI KUMAR ,M. Tech,(Ph.D)

Assoc. Professor

Department of CSE

SITAM

Head of the department

DR.G.VENU MADHAVA RAO, MTech, Ph.D.

Professor & HOD

Department of CSE

SITAM

External Examiner

ACKNOWLEDGEMENT

It is with profound gratitude that we express our deep indebtedness to our guide **Mr.T.ANJI KUMAR ,M.Tech , (Ph.D)**, Associate professor, Department of Computer Science and Engineering , for his valuable advice and technical support throughout our project work. We consider ourselves greatly honored to have obtained a chance of working under him.

We wish to take this opportunity to express my deepest sense of gratitude to **Dr.G.VENU MADHAVA RAO,M.Tech,Ph.d, Head of the Department CSE** for his encouragement and help in our project. We are very thankful to all our faculty members for their cooperation and guidance which have been a significant factor for the accomplishment of this project.

We would also like to thank Prof **D. V. RAMAMURTHY, Principal, SITAM,** and Vizianagaram for his kind consent in doing the course, project, and for incitement towards us

Also, we extend our sincere thanks to the staff and the librarian regarding all their help and support throughout this project. We are also indebted to our parents and friends for their encouragement and for helping us in our endeavors. Last, but not least, we would thank everyone who has contributed to the successful completion of our project.

Sincerely,

D.A.GAYATRI	(19B61A0513)
D.BHANU PRATAP	(19B61A0512)
M.JYOTSNA HARIKA	(19B61A0530)
B.V.S. POOJITHA	(19B61A0505)
G. DURGA NAIDU	(19B61A0519)

DECLARATION

We hereby declare that the project entitled “**Digital Wallet**“ submitted to the fulfillment of the degree of **B.TECH(CSE)** in **Satya Institute of Technology and Management**, affiliated to the **Jawaharlal Nehru Technological University, Kakinada**. This project work in original has not been submitted so far in any part or full for any other university or institute for the award of any degree or diploma

D.A.GAYATRI	(19B61A0513)
D.BHANU PRATAP	(19B61A0512)
M.JYOTSNA HARIKA	(19B61A0530)
B.V.S.POOJITHA	(19B61A0505)
G. DURGA NAIDU	(19B61A0519)

ABSTRACT

A digital wallet is a modern form of financial technology that enables users to make electronic transactions without carrying physical cash or cards. It is a virtual platform that stores payment information, such as credit and debit card details, and facilitates transactions through the internet or a mobile app.

One of the key advantages of using a digital wallet is convenience. Users can make purchases or send money to others with just a few taps on their smartphone or computer. This eliminates the need to carry cash or physically visit a bank to make a transfer. Additionally, many digital wallets offer rewards and discounts to users for making transactions through their platform. Another advantage of using a digital wallet is security. With traditional payment methods, there is a risk of losing cash or having credit card information stolen.

CONTENTS

Chapter 1: INTRODUCTION	1
Chapter 2: LITERATURE SURVEY	2
Chapter 3:SYSTEM ANALYSIS	3
3.1) Existing System	4
3.2) Proposed System	4
3.3) Feasibility Study	4
Chapter 4: SYSTEM REQUIREMENTS SPECIFICATION	
4.1) Functional Requirements	6
4.2) Non -Functional Requirements	6-7
4.3) Hardware Requirements	7
4.4) Software Requirements	7
Chapter 5: SYSTEM DESIGN	
5.1) Uml Diagrams	8
5.1.1) Definition	8
5.1.2) Building Block Of Uml	9-10
5.1.3) Staruml	10-11
5.1.4) For User	12
5.1.5) For Admin	13
Chapter 6: IMPLEMENTATION	
6.1) Technologies Used	14
6.1.1) Windows 7 OS	14
6.1.2) Html	14
6.1.3) Css	14
6.1.4) Js	15
6.1.5) Firebase	15
6.1.6) Vs Code	16
6.2) Sample Code	17-48
Chapter 7: SAMPLE SCREENS	48-54
Chapter 8: SYSTEM TESTING	54-55

8.1) Software Testing	54
8.2) Testing Methodologies	55
Chapter 9: CONCLUSION AND FUTURE ENHANCEMENTS	56
Chapter 10: REFERENCES	57
Chapter 11: BIBLIOGRAPHY	58

1. INTRODUCTION

A digital wallet, also known as an e-wallet, is a virtual platform that allows users to store and manage their payment information, such as credit and debit card details, and make electronic transactions. Digital wallets are typically accessed through a mobile app or website.

One of the key advantages of using a digital wallet is convenience. Users can make purchases or send money to others with just a few taps on their smartphone or computer. This eliminates the need to carry cash or physically visit a bank to make a transfer. Additionally, many digital wallets offer rewards and discounts to users for making transactions through their platform.

Another advantage of using a digital wallet is security. With traditional payment methods, there is a risk of losing cash or having credit card information stolen. Digital wallets use encryption and authentication methods to protect users' personal and financial information.

Digital wallets have also made it easier for people to access financial services. In many developing countries, traditional banking services are often inaccessible or too expensive for the average person. Digital wallets offer a low-cost alternative, allowing people to make transactions, pay bills, and access other financial services from their mobile devices.

However, there are also some concerns surrounding digital wallets. One issue is that they require a reliable internet connection to function properly, which can be a challenge in some areas. Additionally, some digital wallets may charge fees for certain transactions or services, which can be a disadvantage for low-income individuals.

2. LITERATURE SURVEY

A survey on digital wallets can provide valuable insights into the adoption and usage of these systems by consumers. Surveys can help identify trends, preferences, and attitudes towards digital wallets, as well as barriers to adoption and opportunities for improvement. Some key findings from recent surveys on digital wallets include:

Increasing adoption: According to a survey by Statista, the percentage of US adults using digital wallets increased from 16% in 2015 to 29% in 2020. The COVID-19 pandemic has accelerated this trend, with more people turning to contactless payment methods to avoid physical contact.

Security concerns: Despite the benefits of digital wallets, security concerns remain a major barrier to adoption. A survey by Finder found that 30% of respondents were concerned about the security of their personal information when using digital wallets.

Ease of use: One of the main advantages of digital wallets is their ease of use, and this is reflected in consumer preferences. A survey by Javelin Strategy & Research found that 56% of consumers rated ease of use as the most important factor when choosing a digital wallet.

Loyalty programs: Digital wallets are increasingly incorporating loyalty programs and rewards to incentivize usage. A survey by Mercator Advisory Group found that 80% of consumers were interested in loyalty programs tied to digital wallets.

Mobile payment integration: The integration of digital wallets with mobile payment apps is becoming more common, with popular apps like Venmo and PayPal offering digital wallet features. A survey by eMarketer found that 52% of US smartphone users had used a mobile payment app in the past year.

User experience: Another area of research has focused on the user experience of digital wallets. Studies have explored how design factors, such as user interface, ease of use, and aesthetics, influence users' adoption and usage of digital wallets.

Digital currencies: Finally, researchers have investigated the role of digital currencies, such as Bitcoin, in digital wallets.

3.SYSTEM ANALYSIS

System analysis is the process of studying and analyzing a system in order to identify its goals, objectives, requirements, and constraints. It involves breaking down a complex system into smaller, more manageable components and examining how these components interact with one another to achieve the system's objectives. The goal of system analysis is to identify areas for improvement, optimization, and innovation within a system.

The process of system analysis typically involves the following steps:

Defining the problem: The first step in system analysis is to identify the problem or opportunity that the system is intended to address. This may involve gathering information about the current state of the system, identifying stakeholders and their needs, and defining the goals and objectives of the system.

Gathering information: The next step is to gather information about the system, including its current state, its stakeholders, and its requirements. This may involve conducting interviews with stakeholders, reviewing documentation, and analyzing data.

Analyzing the system: Once the information has been gathered, the system analyst can begin to analyze the system. This may involve creating models of the system, such as data flow diagrams, process models, or entity-relationship diagrams. The analyst can use these models to identify areas where the system can be improved or optimized.

Identifying requirements: Based on the analysis of the system, the analyst can identify the requirements that the system must meet in order to achieve its objectives. This may involve creating a requirements document that outlines the functional and non-functional requirements of the system.

Designing the system: Once the requirements have been identified, the analyst can begin to design the system. This may involve creating a system architecture, selecting hardware and software components, and designing the user interface.

Implementing the system: The next step is to implement the system, which may involve writing code, configuring hardware and software components, and testing the system to ensure that it meets the requirements.

3.1 EXISTING SYSTEM

PayPal: PayPal is one of the oldest and most well-known digital wallet systems. It allows users to store their payment information, including credit and debit card details, and make payments online and in-store using their PayPal account.

Google Pay: Google Pay is a digital wallet system developed by Google. It allows users to store their payment information, including credit and debit card details, and make payments using their Google device. Google Pay also supports loyalty programs and offers in-app purchases.

3.2 PROPOSED SYSTEM

- The proposed system can make payments using the Digital wallet secure and fast transactions.
- It helps to track the Money.
- Can send or Request Money.
- It can send money using offline sms services for featured phones.
- Admin panel to monitor users and Activities.
- Security to prevent DOS attack , SQL Injection and API hijacking.

3.3 FEASIBILITY STUDY

Market analysis: This involves researching the market demand for a digital wallet system, identifying potential target audiences, and assessing the competition in the market. The study would need to evaluate whether the digital wallet system would have a significant market opportunity and whether there is a need for the system.

Technical feasibility: The study would need to assess the technical feasibility of the proposed digital wallet system. This would involve identifying the required technology and infrastructure to develop and operate the system, as well as assessing the availability and cost of the necessary resources.

Financial feasibility: The study would need to evaluate the financial feasibility of the digital wallet system. This would involve estimating the costs associated with developing and launching the system, as well as forecasting the revenue that the system is likely to generate. The study would need to consider potential sources of funding, such as investors or loans, and assess the potential return on investment for stakeholders.

Operational feasibility: The study would need to assess the operational feasibility of the digital wallet system. This would involve identifying the resources and personnel required to operate the system, as well as assessing the potential risks and challenges associated with maintaining the system over time.

4 . SYSTEM REQUIREMENTS SPECIFICATION

Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation such as performance engineering requirements, quality standards, or design constraints.

4.1 FUNCTIONAL REQUIREMENTS

- 1) Create user accounts.
- 2) creating user's data in a database.
- 3) Login and Register Pages
- 4) Access database information after login.
- 5) Check the user and owner transactions
- 6) Sending and Receiving of money.
- 7) Track of transaction .
- 8) Log out from the account.
- 9) Offline sms database.

4.2 NON- FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows:

Reliability

The system is more reliable because it was built using web technologies where the majority of devices can access this system.

Performance

This system is developed using the advanced front-end and back-end technologies it will give response to the end user on the client system within very less time.

Supportability

The system is designed to be cross platform supportable. The system is supported on a wide range of hardware and any software platform, which has a browser.

Security

This security assures that all data inside the system or its part will be protected against malware attacks or unauthorized access. If you want to protect the admin panel from unauthorized access, you would define the login flow and different user roles as system behavior or user actions and contain security rules for the database , can prevent unwanted requests.

4.3 HARDWARE REQUIREMENTS

- System : Pentium,i3 2.53 GHz.
- Hard Disk : 40 GB.
- Ram : 4 GB.

4.4 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 or later
- Language : HTML, CSS and JS
- Database : Firebase
- Server : Firebase
- Editor : VS code

5. SYSTEM DESIGN

5.1 UML DIAGRAMS

UML is a method for describing the system architecture in detail using the blueprint. UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. UML is a very important part of developing objects oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

5.1.1 Definition

UML is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of the software system.

UML is a Language

It will provide vocabulary and rules for communications and function on conceptual and physical representation. So it is a modeling language.

UML Specifying

Specifying means building models that are precise, unambiguous and complete. In particular, the UML addresses the specification of all the important analysis, design and implementation decisions that must be made in developing and displaying a software intensive system.

UML Visualization

The UML includes both graphical and textual representation. It makes it easy to visualize the system and for better understanding.

UML Constructing

UML models can be directly connected to a variety of programming languages and it is sufficiently expressive and free from any ambiguity to permit the direct execution of models.

Uses of UML

The UML is intended primarily for software intensive systems. It has been used effectively for such domain as

- o Enterprise Information System
- o Banking and Financial Services
- o Telecommunications
- o Transportation
- o Defense
- o Retails
- o Medical Electronics
- o Scientific Fields

5.1.2 Building Blocks of UML

The vocabulary of the UML encompasses 3 kinds of building blocks

- I. Things
- II. Relationships
- III. Diagrams

Things

Things are the data abstractions that are first class citizens in a model. Things are of 4 types Structural Things, Behavioral Things ,Grouping Things, An notational Things

Relationships

Relationships tie things together. Relationships in the UML are Dependency, Association, Generalization, Specialization

There are two types of diagrams, they are: Structural and Behavioral Diagrams

Structural Diagrams

The UML's four structural diagrams exist to visualize, specify, construct and document the static aspects of a system. Icon View the static parts of a system using one of the following diagrams. Structural diagrams consist of Class Diagram, Object Diagram, Component Diagram, and Deployment Diagram.

Behavioral Diagrams

The UML's five behavioral diagrams are used to visualize, specify, construct, and document the

dynamic aspects of a system. The UML's behavioral diagrams are roughly organized around the major ways which can model the dynamics of a system. Behavioral diagrams consist of Use case Diagram, Sequence Diagram, Collaboration Diagram, State chart Diagram, Activity Diagram.

Use Case diagram

A use case is a set of scenarios that describe an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

5.1.3 StarUML

StarUML is a sophisticated software modeler aimed to support *agile* and *concise* modeling. **StarUML** is a UML tool by MKLab. The software was licensed under a modified version of GNU GPL until 2014, when a rewritten version 2.0.0 was released for beta testing under a proprietary license. After being abandoned for some time, the project had a revival to move from Delphi to Java/Eclipse and then stopped again. In 2014, a rewritten version was released as proprietary software. The stated goal of the project was to replace larger, commercial applications such as Rational Rose and Borland Together.

StarUML supports most of the diagram types specified in UML 2.0. It is currently missing timing and interaction overview diagrams. StarUML was written in Delphi, which is one of the reasons why it was abandoned for a long time. Since December 2005 StarUML was not updated anymore, although some external modules were updated. Currently the newest version of StarUML by the original authors is available for download under the handle "StarUML 2". The public beta is available, although not under the GPL license.

The main targets of users are:

- Agile and small development teams
- Professional persons
- Educational institutes.

The key features of StarUML are:

- Multi-platform support (MacOS, Windows and Linux)
- UML 2.x standard compliant
- Entity-Relationship diagram (ERD)
- Data-flow diagram (DFD)
- Flowchart diagram
- Retina (High-DPI) display support
- Model-driven development
- Open APIs
- Various third-party extensions
- Asynchronous model validation

USE CASE DIAGRAM

5.1.4 For User:

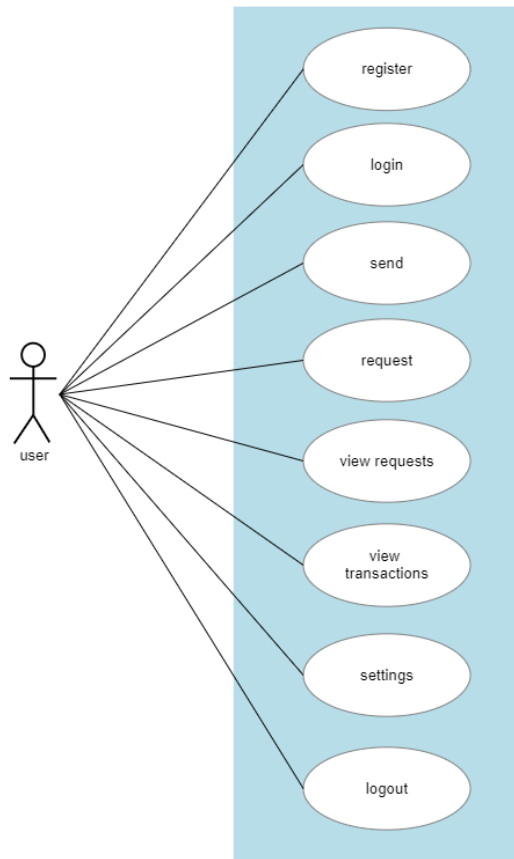
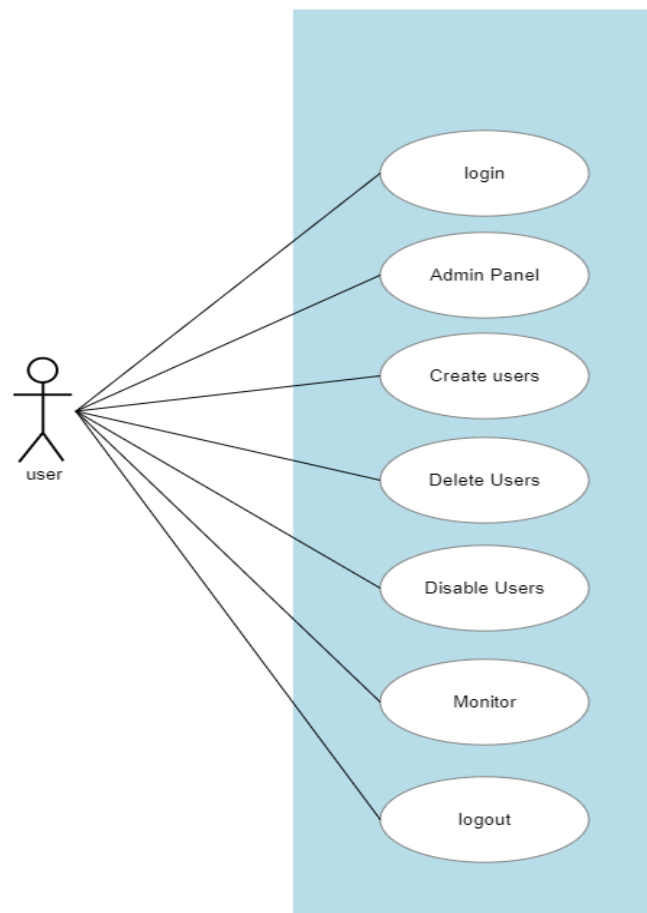


Fig 5.1: Use case Diagram

In this project users have the functionality of creating a new account by giving the input of Email, Phone number and password at register and can use the same credentials to login to account. The home contains Wallet information and user name along with phone number. Home page contains a Send, Request and Deposit button which can be accessed by clicking. Each page has a bottom bar menu for easy access of tabs in the project i.e Home Screen, Request Screen, Transactions History and Setting screen.

5.1.5 For Admin:



The admin page contains the monitor system which is able to control the project information, prevent attacks and can manage the user activity like creating, updating and disabling them using the dashboard. Admin has the special feature to handle and can login and logout using the admin credential.

6. IMPLEMENTATION

6.1 TECHNOLOGIES USED

6.1.1 WINDOWS 7 OPERATING SYSTEM

Windows 7 is an operating system that was produced by Microsoft as part of the Windows NT family of operating systems. It was released to manufacturing on July 22, 2009, and became generally available on October 22, that year, less than three years after the release of its predecessor, Windows Vista. Windows 7's server counterpart, Windows Server 2008 R2, was released at the same time. Microsoft ended mainstream support for Windows 7 on January 13, 2015 and extended support ended on January 14, 2020.

6.1.2 HTML:

HTML, or Hypertext Markup Language, is a programming language used for creating web pages and other information that can be displayed in a web browser. HTML is used to structure content on the web page, including text, images, videos, and links to other pages or resources.

HTML consists of tags, which are used to define the structure and content of the page. For example, the `<h1>` tag is used to define a heading, while the `<p>` tag is used to define a paragraph of text. HTML also allows for the use of attributes, which provide additional information about the element being defined.

HTML is a foundational language for web development, and it is often used in conjunction with other web technologies such as CSS (Cascading Style Sheets) and JavaScript to create interactive and visually appealing web pages.

6.1.3 CSS :

CSS, or Cascading Style Sheets, is a web technology used to style and format the presentation of HTML and XML documents. CSS is used to define the visual appearance of a web page, including colors, fonts, layout, and other visual effects.

CSS works by defining rules, which are applied to elements on a web page. These rules specify how an element should appear, such as the font size, color, and spacing. CSS can be used to apply styles to specific elements, or to groups of elements, using selectors.

CSS can be written directly into HTML documents using the `<style>` tag, or it can be stored in separate CSS files and linked to the HTML document using the `<link>` tag. Separating CSS from HTML allows for easier maintenance and management of web pages.

CSS is an important tool for web developers, as it enables the creation of visually appealing and responsive websites. It is often used in conjunction with other web technologies such as HTML and JavaScript to create dynamic and interactive web pages.

6.1.4 JS:

JavaScript (JS) is a programming language used for creating interactive and dynamic web pages. It is a high-level, object-oriented language that is primarily used in web development.

JavaScript works by allowing developers to add interactive elements to a webpage, such as animations, dropdown menus, and pop-ups. It can also be used for more complex applications such as online games, data visualization, and real-time communication.

JS is commonly used in conjunction with HTML and CSS to create responsive and interactive web pages. It can be included directly in the HTML code or stored in separate files and linked to the HTML document. JS can also interact with server-side technologies such as PHP, Node.js, and Ruby on Rails.

JS has many powerful frameworks and libraries that can be used to create complex web applications. Some popular JS frameworks include React, Angular, and Vue.js, while popular JS libraries include jQuery and D3.js.

Overall, JavaScript is a versatile and essential tool for web developers, and its continued development and adoption has enabled the creation of more dynamic and interactive web experiences.

6.1.5 Firebase: Firebase is a mobile and web application development platform that provides a range of backend services and tools for building and deploying applications. It is a

cloud-based platform that is owned by Google and offers a range of features, including authentication, real-time database, cloud storage, cloud functions, and hosting.

Firebase's authentication service provides secure and easy-to-use authentication for users, allowing developers to easily add login and authentication features to their applications. The real-time database service provides a NoSQL database that can be used to store and sync data in real time between clients and servers. Cloud storage provides secure and scalable storage for files and media, while cloud functions allow developers to run code in response to events triggered by their applications.

Firebase also provides hosting for web applications, allowing developers to deploy and host their applications with ease. Additionally, Firebase provides tools for testing and analytics, enabling developers to monitor their applications and gain insights into user behavior.

Firebase is designed to be easy to use, even for developers who have little experience with backend development. It also provides extensive documentation and support, making it easy for developers to learn and use the platform.

6.1.6 VSCode:

VSCode, short for Visual Studio Code, is a popular and widely used source-code editor developed by Microsoft. It is a free, open-source tool that is available for Windows, macOS, and Linux.

VSCode has a range of features that make it a popular choice for developers, including support for multiple programming languages, code highlighting, auto-completion, debugging, and extensions. It also has a built-in terminal that allows developers to run commands and scripts directly within the editor.

One of the most powerful features of VSCode is its extensibility. It has a large and active community of developers who create and maintain extensions, which can be used to enhance the functionality of the editor. There are thousands of extensions available for VSCode, covering a wide range of use cases and programming languages.

VSCode also has a highly customizable user interface, allowing developers to customize their workspace with their preferred layout, themes, and font settings. It also provides

a range of productivity features, such as keyboard shortcuts and code snippets, which can help developers work more efficiently.

Overall, VSCode is a highly popular and powerful source-code editor that is widely used by developers around the world. Its extensibility, customization options, and productivity features make it a popular choice for developers working in a wide range of programming languages and environments.

6.2 Sample Code:

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="manifest" href="__manifest.json">
  <link href="https://fonts.googleapis.com/css2?family=Viga&display=swap"
rel="stylesheet">
  <script src="base.js"></script>
  <title>Document</title>
  <style>
    #preload{
      background: rgb(255, 255, 255) url("bhanu.gif") no-repeat center center;
      height: 100vh;
      width: 100%;
      position: fixed;
      z-index: 100;
    }
    *{
      padding: 0;
```



```

margin: 0;
overflow: hidden;
overflow-y: visible;
}
.hi {
color: #090909;
padding: 0.1em 1.7em;
margin-left: -36%;
width: 156%;
max-width: 156%;
color: white;

}

button {
color: #090909;
padding: 0.7em 1.7em;
font-size: 18px;
border-radius: 0.5em;
background: #41466a;
border: 1px solid #e8e8e8;
transition: all .3s;
box-shadow: 6px 6px 12px #c5c5c5,
            -6px -6px 12px #ffffff;
margin-left: 25%;
width: 50%;
color: white;
}

button:active {

```

```

color: #666;
box-shadow: inset 4px 4px 12px #c5c5c5,
            inset -4px -4px 12px #ffffff;
}
h3{
padding: 0.7em 1.7em;
margin-left: 10%;
width: 100%;
}
</style>
</head>
<body>
  <div id="preload"></div>
  
  <a href="login.html" > <button>Log in</button></a><br>
  <br>
  <h3>Don't Have Account?<a href="register.html">Sign Up</a></h3>
</body>
<script>
  AddtoHome("2000", "once");
  var load = document.getElementById("preload");
  setTimeout(() => {
    load.setAttribute("style", "pointer-events: none; opacity: 0; transition: 0.5s ease-in-out;");
    setTimeout(() => {
      load.setAttribute("style", "display: none;")
    }, 1000);
  }, 450);
</script>
</html>

```

Home.html

```
<html>
  <head>
    <title>Log In</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link
      rel="stylesheet"
      href="https://fonts.googleapis.com/icon?family=Material+Icons"
    />
    <link href="https://fonts.googleapis.com/css2?family=Viga&display=swap"
rel="stylesheet">
    <link rel="manifest" href="__manifest.json">
    <link rel="stylesheet" href="main.css">
    <style>
      #preload{
        background: rgb(255, 255, 255) url("bhanu.gif") no-repeat center center;
        height: 100vh;
        width: 100%;
        position: fixed;
        z-index: 100;
      }
      input,
      textarea,
      button,
      select,
      a {
        -webkit-tap-highlight-color: transparent;
```

```

    }

</style>
</head>
<body>
    <div id="preload"></div>
    <div id="topbar">
        <h2><center><div id="username">Welcome!
User</div></center></h2><br><br>
        <br><center><span id="walletbalance">₹ ...</span></center>
    </div>
    <div class="appTopMenu">
        <a href="send.html" class="item active">
            <div class="col">
                
            </div>
        </a>
        <a href="recive.html" class="item">
            <div class="col">
                
            </div>
        </a>
        <a href="checkout.html" class="item">
            <div class="col">
            </div>
        </a>
    </div><br>
    <div class="appBottomMenu">
        <a href="home.html" class="item active">

```

```

    <div class="col">
      
    </div>
  </a>
  <a href="request.html" class="item">
    <div class="col">
    </div>
  </a>
  <a href="Trans.html" class="item">
    <div class="col">
    </div>
  </a>
  <a href="setting.html" class="item">
    <div class="col">
    </div>
  </a>
</div>
<script type="module">

```

```

  // Import the functions you need from the SDKs you need
  import { initializeApp } from
  "https://www.gstatic.com/firebasejs/9.17.1/firebase-app.js";

  import { getAuth, onAuthStateChanged, updateProfile } from
  "https://www.gstatic.com/firebasejs/9.17.1/firebase-auth.js";

  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration

```

```

// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyBaLdcoeVZYBOGXwE3OaWknh1lbhp9_IS0",
  authDomain: "walletdigitalcoin.firebaseio.com",
  databaseURL:
"https://walletdigitalcoin-default-rtdb.asia-southeast1.firebaseio.com",
  projectId: "walletdigitalcoin",
  storageBucket: "walletdigitalcoin.appspot.com",
  messagingSenderId: "797876183390",
  appId: "1:797876183390:web:f896a9b4c522e33759efcf",
  measurementId: "G-J42NENRFPW"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
import { doc, getDoc, setDoc, getFirestore, addDoc
,collection,updateDoc,increment} from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-firestore.js";
import {getDatabase, ref, get, set,child, update, remove}
from "https://www.gstatic.com/firebasejs/9.17.1/firebase-database.js";
const auth = getAuth();
const user = auth.currentUser;
var uname = "
onAuthStateChanged(auth, async (user) => {
  if (user) {
    // User is signed in, see docs for a list of available properties
    // https://firebase.google.com/docs/reference/js/firebase.User
    const uid = user.uid;

```

```

uname = user.displayName;
var email_id = user.email;
var email_verified = user.emailVerified;
if(email_verified)
{
const db =getFirestore();

var walletbal = document.getElementById("walletbalance");
var username = document.getElementById("username");
var urname = ""
var dref = doc(db,'users',uname);

const docSnap = await getDoc(dref);

if(docSnap.exists()){
var a = docSnap.data()
walletbal.innerHTML = "₹ "+a.wallet;
username.innerHTML= "welcome "+a.name;
}
else{
alert(0);
}
}
else{
window.location='verfiy.html';
}

```

```

    }
    else{
        window.location='login.html';
    }
});

```

```

var load = document.getElementById("preload");
setTimeout(() => {
    load.setAttribute("style", "pointer-events: none; opacity: 0; transition: 0.2s
ease-in-out;");
    setTimeout(() => {
        load.setAttribute("style", "display: none;")
    }, 1000);
}, 450);

```

```

</script>

```

```

</body>

```

```

</html>

```

Checkout.html

```

<!DOCTYPE html>

```

```

<html lang="en">

```

```

<head>

```

```

    <meta charset="UTF-8">

```

```

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

```

```

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

    <title>Checkout</title>

```

```

    <link rel="stylesheet" href="main.css">

```



```

<style>
  #preload{
    background: rgb(255, 255, 255) url("bhanu.gif") no-repeat center center;
    height: 100vh;
    width: 100%;
    position: fixed;
    z-index: 100;
  }
  #back{
    background-color: black;
    padding: 5%;
    margin: 10%;
  }
</style>
</head>
<body>
  <div id="preload"></div>
  <Div class="inp">
    Amount<br><br>
    <input type="number" id="dep" placeholder="Enter the Amount"><br><br>
    <button id="rzp-button1">Pay</button>
  </Div>
</body>
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
<script src="https://checkout.razorpay.com/v1/checkout.js"></script>
<script type="module">

```

```

// Import the functions you need from the SDKs you need
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.17.1/firebase-app.js";
import { getAuth, onAuthStateChanged } from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-auth.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyBaLdcoeVZYBOGXwE3OaWknh1lbhp9_IS0",
authDomain: "walletdigitalcoin.firebaseio.com",
databaseURL: "https://walletdigitalcoin-default-rtdb.asia-southeast1.firebaseio.com",
projectId: "walletdigitalcoin",
storageBucket: "walletdigitalcoin.appspot.com",
messagingSenderId: "797876183390",
appId: "1:797876183390:web:f896a9b4c522e33759efcf",
measurementId: "G-J42NENRFPW"
};

var amot = document.getElementById("dep");
var numb = document.getElementById("p_no");
var amount_order = 0;
var phone_number;
// Initialize Firebase
const app = initializeApp(firebaseConfig);
import { getDatabase, ref, get, set, child, update, remove }
from "https://www.gstatic.com/firebasejs/9.17.1/firebase-database.js";

```

```

import { doc, setDoc,getFirestore,addDoc ,collection,updateDoc,increment,serverTimestamp}
from "https://www.gstatic.com/firebasejs/9.17.1/firebase-firestore.js";

// const db =getDatabase();
const db = getFirestore(app);


const auth = getAuth();
var uname=""
onAuthStateChanged(auth, (user) => {
if (user) {
const uid = user.uid;
uname = user.displayName;
}
else{
window.location='login.html';
}
});
document.getElementById('rzp-button1').onclick = function(e){
amount_order = parseInt(amot.value);
var options = {
"key": 'rzp_test_RCq3JwcqSp6LEL',
"amount": amount_order*100,
"currency": "INR",
"description": "Sample Order",
"handler": function(response) {
updateDoc(doc(db, "users", uname), {

```

```

wallet: increment(amount_order)
});
Swal.fire(
  'Deposit Success',
  "",
  'success'
)
},
"prefill": {
  "name": "Bhanu",
  "email": "dadibhanupratap@gmail.com"
}
};
var rzp2 = new Razorpay(options);
rzp2.open();
e.preventDefault();
}
var load = document.getElementById("preload");
setTimeout(() => {
  load.setAttribute("style", "pointer-events: none; opacity: 0; transition: 0.5s ease-in-out;");
  setTimeout(() => {
    load.setAttribute("style", "display: none;")
  }, 1000);
}, 450);
</script>
</html>

```

Transactions.html

```
<html>
```

```

<head>
  <title>Log In</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://fonts.googleapis.com/css2?family=Viga&display=swap"
rel="stylesheet">
  <style>
    *{
      padding: 0;
      margin: 0;
    }
    #transactions{
      padding: 10px;
      border-radius:3%;
      width: 90%;
      margin-left: 2%;
      color: white;
      background-color: black;
    }
    a{
      text-decoration: none;
      color: white;
    }
    body{
      padding-bottom: 15%;
      font-family: "Viga", serif;
    }
    .appBottomMenu{
      background-color:white;

```

```
overflow: hidden;
position: fixed;
bottom: 0;
width: 100%;
padding-top: 1em;
}
```

```
.appBottomMenu a{
    float: left;
display: block;
color: #f2f2f2;
text-align: center;
width: 25%;
text-decoration: none;
font-size: 17px;
margin-bottom: 1em;
}
```

```
.box1 {
    float:left;
    width: 33.33%;
    padding: 10px;
    padding-left: 1em;
    margin-top: 3%;
```

```
}
```

```
#box2{
    float:left;
```

```

        width: 33.33%;
        padding: 10px;
        margin-top: 3%;
    }
    #box3 {
        float: left;
        width: 33.33%;
        padding: 10px;
        margin-top: 3%;
        text-align: center;
    }
    #tnx {
        content: "";
        clear: both;
        display: flex;
    }
    .bhanu {
        padding: 2em;
    }
    #preload {
        background: rgb(255, 255, 255) url("bhanu.gif") no-repeat center center;
        height: 100vh;
        width: 100%;
        position: fixed;
        z-index: 100;
    }
</style>
</head>

```

```

<body>
  <div id="preload"></div>
  <center><h1>Transactions</h1></center><br><br>
  <div id="bhanu">
  </div>
  <div class="appBottomMenu">
    <a href="home.html" class="item active">
      <div class="col">
        
      </div>
    </a>
    <a href="request.html" class="item">
      <div class="col">
      </div>
    </a>
    <a href="Trans.html" class="item">
      <div class="col">
      </div>
    </a>
    <a href="setting.html" class="item">
      <div class="col">
      </div>
    </a>
  </div>

  <script type="module">
    // Import the functions you need from the SDKs you need

```



```

import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-app.js";

import { getAuth, onAuthStateChanged, updateProfile } from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-auth.js";

// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries


// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyBaLdcoeVZYBOGXwE3OaWknh1lbhp9_IS0",
  authDomain: "walletdigitalcoin.firebaseio.com",
  databaseURL:
"https://walletdigitalcoin-default-rtdb.asia-southeast1.firebaseio.com",
  projectId: "walletdigitalcoin",
  storageBucket: "walletdigitalcoin.appspot.com",
  messagingSenderId: "797876183390",
  appId: "1:797876183390:web:f896a9b4c522e33759efcf",
  measurementId: "G-J42NENRFPW"
};


// Initialize Firebase
const app = initializeApp(firebaseConfig);
import { getFirestore, setDoc, addDoc, doc, collection
,getDoc, getDocs, updateDoc, where, query, orderBy } from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-firestore.js";

const auth = getAuth();
const user = auth.currentUser;

```

```

    var uname = "
onAuthStateChanged(auth, async (user) => {
  if (user) {
    uname = user.displayName;
    const db = getFirestore(app);
    const q = query(collection(db,
"users",uname,"Transactions"),orderBy("time","desc"),where("time", "!=", false));
    const querySnapshot = await getDocs(q);
    var data = [];
    querySnapshot.forEach((doc) => {
      AddItemsToTable(doc.data().from,doc.data().to,doc.data().amount,doc.id,doc.data().tim
e);
    });
  }
});

```

```

function AddItemsToTable(from,to,amounts,transids,time){
  let lin = document.createElement("a");
  let trans = document.createElement("div");
  let details = document.createElement("div");
  lin.appendChild(details);
  let phone = document.createElement("div");
  lin.appendChild(phone);
  let amount = document.createElement("div");
  lin.appendChild(amount);
  trans.appendChild(lin);

```

```

lin.id = "tnx";
trans.id = 'transactions';
details.className = 'box1';
phone.id = 'box2';
amount.id = 'box3';
if(amounts<0){
details.innerHTML = "Send To";
phone.innerHTML = to;
amount.style = "color:red";
amount.innerHTML =
"<span>$&nbsp</span>"+amounts+"<span>&nbspDR</span>";
}
else{
details.innerHTML = "Recived From";
phone.innerHTML = from;
amount.style = "color:green";
amount.innerHTML =
"<span>$&nbsp</span>"+amounts+"<span>&nbspCR</span>";
}
bhanu.append(trans);
bhanu.append(document.createElement("br"));
bhanu.append(document.createElement("br"));
}
var stdio = 0;
var bh = document.getElementById('bhanu');
function AddAllItemsToTable(theStudents){
    theStudents.forEach(element=>{

```

```

AddItemsToTable(element.from,element.to,element.transid,element.id)
    });
}
function GetAllDataOnce(){
    const dbRef = ref(db);

    get(child(dbRef,"Users/Transations"))
    .then((Snapshot)=>{
        var students =[];

        Snapshot.forEach(childSnapshot=>{
            students.push(childSnapshot.val());
        });
        AddAllItemsToTable(students);
    })
}
// window.onload = GetAllDataOnce;
var load = document.getElementById("preload");
setTimeout(() => {
    load.setAttribute("style", "pointer-events: none; opacity: 0; transition: 0.5s
ease-in-out;");
    setTimeout(() => {
        load.setAttribute("style", "display: none;")
    }, 1000);
}, 450);
</script>

```

```
</body>
</html>
```

Manifest.json:

```
{
  "name": "Digital Wallet",
  "short_name": "Digital wallet",
  "start_url": "index.html",
  "display": "standalone",
  "scope": "/",
  "background_color": "#000000",
  "theme_color": "#000000",
  "orientation": "portrait",
  "icons": [
    {
      "src": "wallet.png",
      "type": "image/png",
      "sizes": "512x512",
      "purpose": "any maskable"
    },
    {
      "src": "wallet128.png",
      "type": "image/png",
      "sizes": "128x128",
      "purpose": "any maskable"
    }
  ]
}
```

Service-worker.js

```

var VERSION = 2.1

// Cache name
var CACHE_NAME = 'cache-version-' + VERSION;

// Files
var REQUIRED_FILES = [
  'index.html'
];

self.addEventListener('install', function (event) {
  // Perform install step: loading each required file into cache
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(function (cache) {
        // Add all offline dependencies to the cache
        return cache.addAll(REQUIRED_FILES);
      })
      .then(function () {
        return self.skipWaiting();
      })
  );
});

```

```

self.addEventListener('fetch', function (event) {
  event.respondWith(
    caches.match(event.request)
      .then(function (response) {
        // Cache hit - return the response from the cached version
        if (response) {
          return response;
        }
        // Not in cache - return the result from the live server
        // `fetch` is essentially a "fallback"
        return fetch(event.request);
      })
  );
});

```

```

self.addEventListener('activate', function (event) {
  // Calling claim() to force a "controllerchange" event on navigator.serviceWorker
  event.waitUntil(self.clients.claim());
});

```

Settings.html

```

<html>
  <head>
    <title>Log In</title>

```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="https://fonts.googleapis.com/css2?family=Viga&display=swap"
rel="stylesheet">
<style>
  #preload{
    background: rgb(255, 255, 255) url("bhanu.gif") no-repeat center center;
    height: 100vh;
    width: 100%;
    position: fixed;
    z-index: 100;
  }
  *{
    padding: 0;
    margin: 0;
  }
  #profile img{
    height: 120px;
    width: 120px;
    margin-top: 10%;
    margin-left: 35%;
  }

  button{
    padding: 20px;

    text-decoration: none;
    color: white;
    background-color: rgb(0, 0, 0);
```



```

        width: 90%;
        margin-left: 5%;
        margin-top: 10px;
        font-family: "Viga", serif;
    }
    .appBottomMenu{
        background-color:white;
overflow: hidden;
position: fixed;
bottom: 0;
width: 100%;
padding-top: 1em;
    }
    body{
        font-family: "Viga", serif;
    }
    .appBottomMenu a{
        float: left;
display: block;
color: #f2f2f2;
text-align: center;
width: 25%;
text-decoration: none;
font-size: 17px;
margin-bottom: 1em;
    }
    a{
        text-decoration: none;

```

```

        color: #f2f2f2;
    }
    #list{
        margin-top: 4em;
    }
</style>
</head>
<body>
    <div id="preload"></div>
<div id="profile">
    
    <br>
    <div><center><h2 id="name">Username</h2></center></div>
</div>
<div id="list">
    <button><a href="updateusername.html">Update Username </a></button>
    <button>Update PIN </button>
    <button>FAQ</button>
    <button id="logout">Log Out</button>
</div>
<div class="appBottomMenu">
    <a href="home.html" class="item active">
        <div class="col">
            
        </div>
    </a>
    <a href="request.html" class="item">
        <div class="col">

```

```

        
    </div>
</a>
<a href="Trans.html" class="item">
    <div class="col">
    </div>
</a>
<a href="setting.html" class="item">
    <div class="col">
    </div>
</a>
</div>
</body>
<script type="module">
    const firebaseConfig = {
        apiKey: "AIzaSyBaLdcoeVZYBOGXwE3OaWknh1lbhp9_IS0",
        authDomain: "walletdigitalcoin.firebaseio.com",
        databaseURL:
"https://walletdigitalcoin-default-rtdb.asia-southeast1.firebaseio.com",
        projectId: "walletdigitalcoin",
        storageBucket: "walletdigitalcoin.appspot.com",
        messagingSenderId: "797876183390",
        appId: "1:797876183390:web:c8bb2345533b200059efcf",
        measurementId: "G-1GE4VP57XD"
    };
    import { initializeApp } from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-app.js";

```

```

import {getAuth,signOut,onAuthStateChanged } from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-auth.js";

import { doc, getDoc,setDoc,getFirestore,addDoc
,collection,updateDoc,increment} from
"https://www.gstatic.com/firebasejs/9.17.1/firebase-firestore.js";


const app = initializeApp(firebaseConfig);
const auth = getAuth();
function logoff(){
  signOut(auth).then(() => {
    console.log("Logout");
    window.location='login.html';
  }).catch((error) => {
    alert("error");
  });
}
const user = auth.currentUser;
var uname = "
onAuthStateChanged(auth, async (user) => {

  if (user) {
    // User is signed in, see docs for a list of available properties
    // https://firebase.google.com/docs/reference/js/firebase.User
    uname = user.displayName;
    var email_id = user.email;
    var email_verified = user.emailVerified;
    const db =getFirestore();
    var username = document.getElementById("name");
  }
}

```

```

var uname = ""
var dref = doc(db,'users',uname);

const docSnap = await getDoc(dref);

if(docSnap.exists()){
    var a = docSnap.data()
    username.innerHTML = a.phonenumber;
}
else{
    alert(0);
}
}
else{
    window.location='login.html';
}
});

logout.addEventListener('click',logoff);
var load = document.getElementById("preload");
setTimeout(() => {
    load.setAttribute("style", "pointer-events: none; opacity: 0; transition: 0.5s
ease-in-out;");
    setTimeout(() => {
        load.setAttribute("style", "display: none;")
    }, 1000);
}, 450);
</script>

```

</html>

404.html

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Page Not Found</title>

<style media="screen">

body { background: #ECEFF1; color: rgba(0,0,0,0.87); font-family: Roboto, Helvetica, Arial, sans-serif; margin: 0; padding: 0; }

#message { background: white; max-width: 360px; margin: 100px auto 16px; padding: 32px 24px 16px; border-radius: 3px; }

#message h3 { color: #888; font-weight: normal; font-size: 16px; margin: 16px 0 12px; }

#message h2 { color: #ffa100; font-weight: bold; font-size: 16px; margin: 0 0 8px; }

#message h1 { font-size: 22px; font-weight: 300; color: rgba(0,0,0,0.6); margin: 0 0 16px; }

#message p { line-height: 140%; margin: 16px 0 24px; font-size: 14px; }

#message a { display: block; text-align: center; background: #039be5; text-transform: uppercase; text-decoration: none; color: white; padding: 16px; border-radius: 4px; }

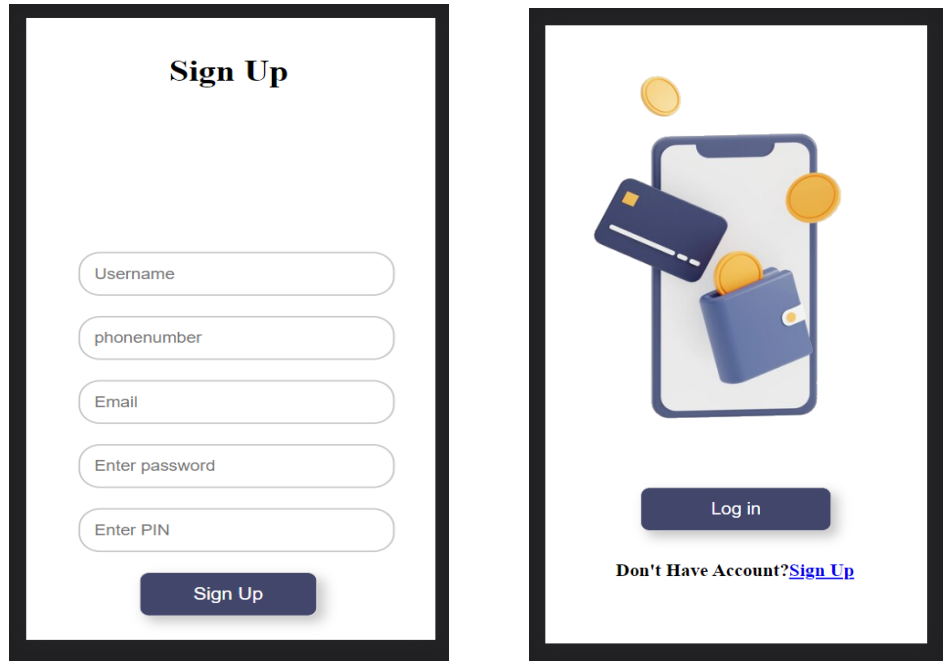
#message, #message a { box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24); }

#load { color: rgba(0,0,0,0.4); text-align: center; font-size: 13px; }

@media (max-width: 600px) {

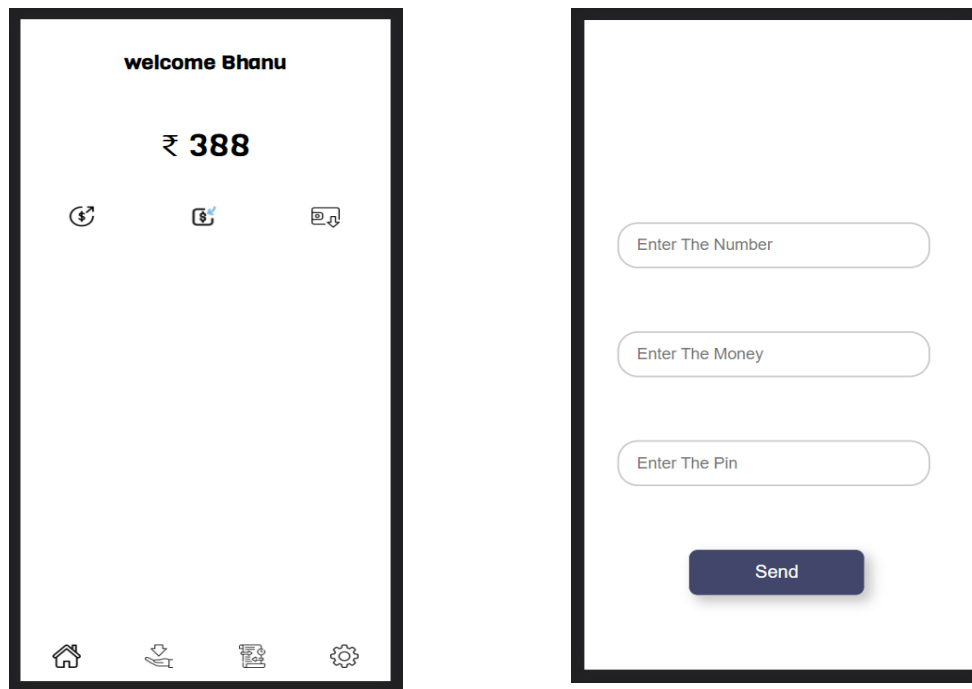
```
    body, #message { margin-top: 0; background: white; box-shadow: none; }
    body { border-top: 16px solid #ffa100; }
  }
</style>
</head>
<body>
  <div id="message">
    <h2>404</h2>
    <h1>Page Not Found</h1>
    <p>The Page will released with new version</p>
    <h3>Why am I seeing this?</h3>
  </div>
</body>
</html>
```

7. SAMPLE SCREENS



The image displays two mobile app screens. The left screen is the 'Sign Up' form, featuring a title 'Sign Up' at the top. Below it are five input fields: 'Username', 'phonenumber', 'Email', 'Enter password', and 'Enter PIN'. A dark blue 'Sign Up' button is positioned at the bottom. The right screen is the 'Login' form, featuring an illustration of a smartphone with a wallet and coins. Below the illustration is a dark blue 'Log in' button. At the bottom, there is a link that says 'Don't Have Account? [Sign Up](#)'.

Fig 7.1: Sign UP and Login forms



The image displays two mobile app screens. The left screen is the 'Home' page, featuring a greeting 'welcome Bhanu' at the top. Below it is a large display of '₹ 388'. There are three small icons below the balance: a dollar sign with a refresh symbol, a wallet icon, and a document icon. At the bottom is a navigation bar with four icons: a home icon, a wallet icon, a document icon, and a settings icon. The right screen is a form for sending money, featuring three input fields: 'Enter The Number', 'Enter The Money', and 'Enter The Pin'. A dark blue 'Send' button is positioned at the bottom.

Fig 7.2: Home page

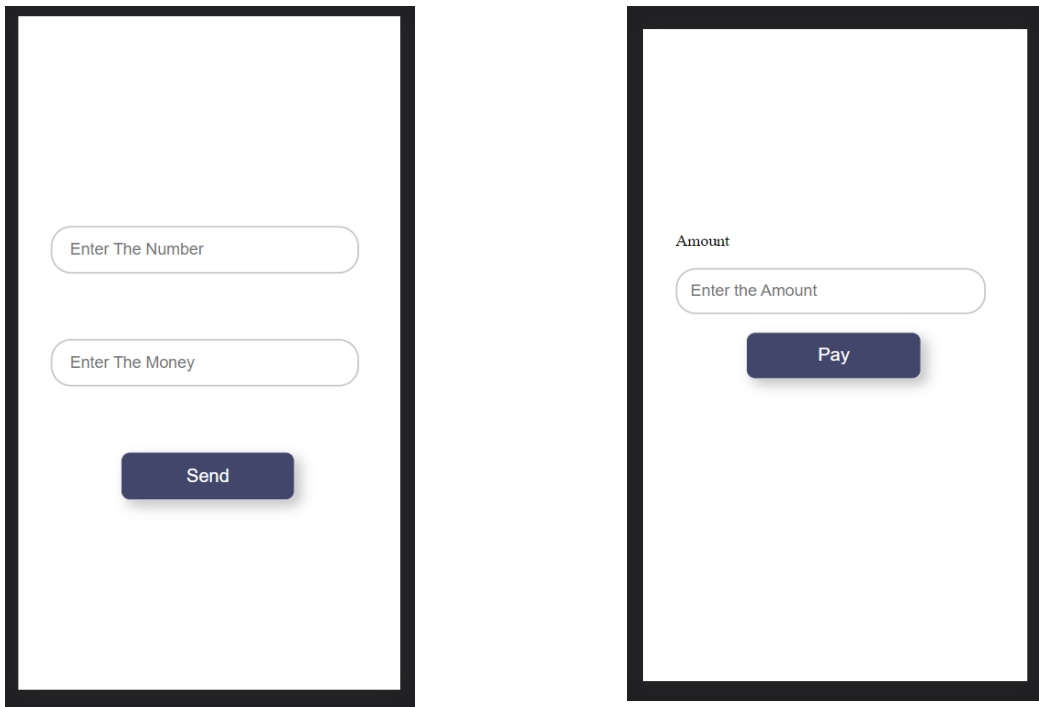


Fig 7.3: Sending and Request pages

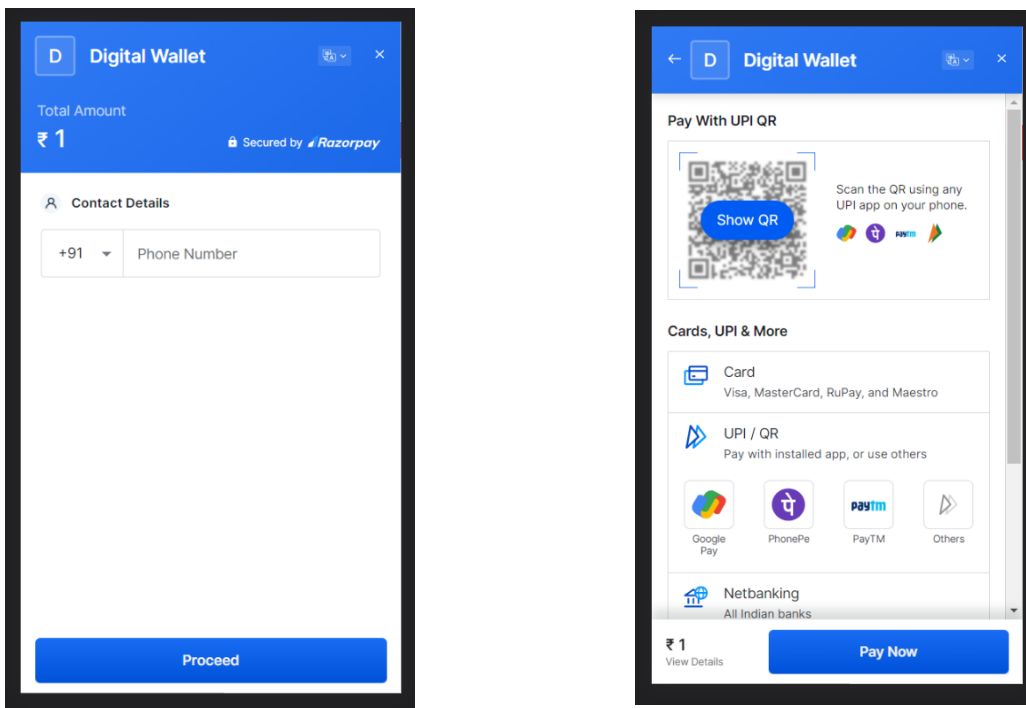


Fig 7.4 Payment Processes

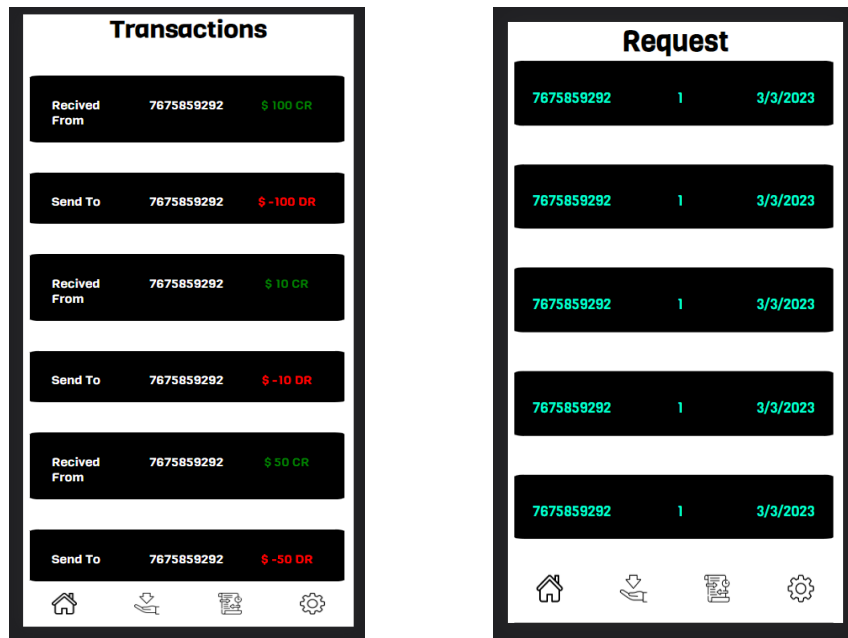


Fig 7.5: Transaction and Request

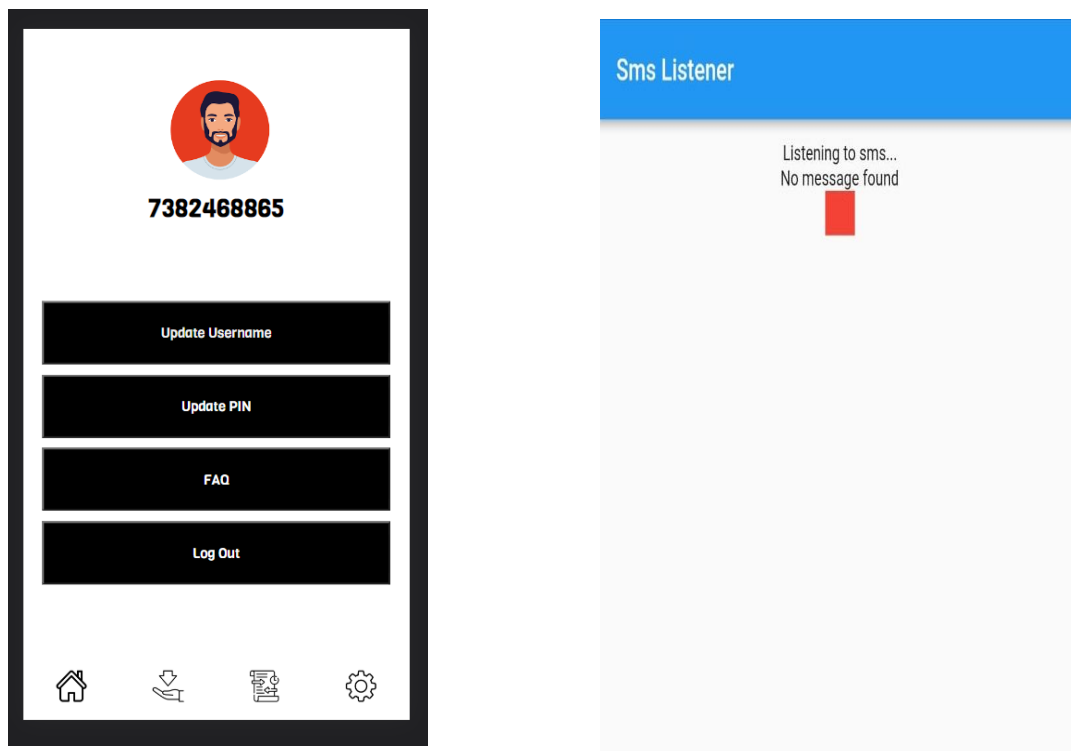


Fig 7.6: Sending SMS

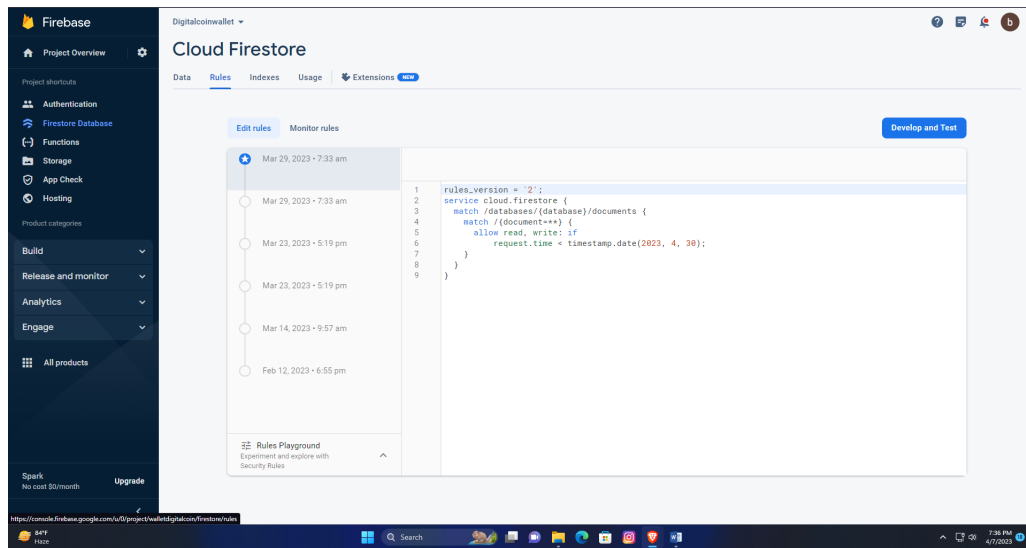


Fig 7.7: Cloud Firestore

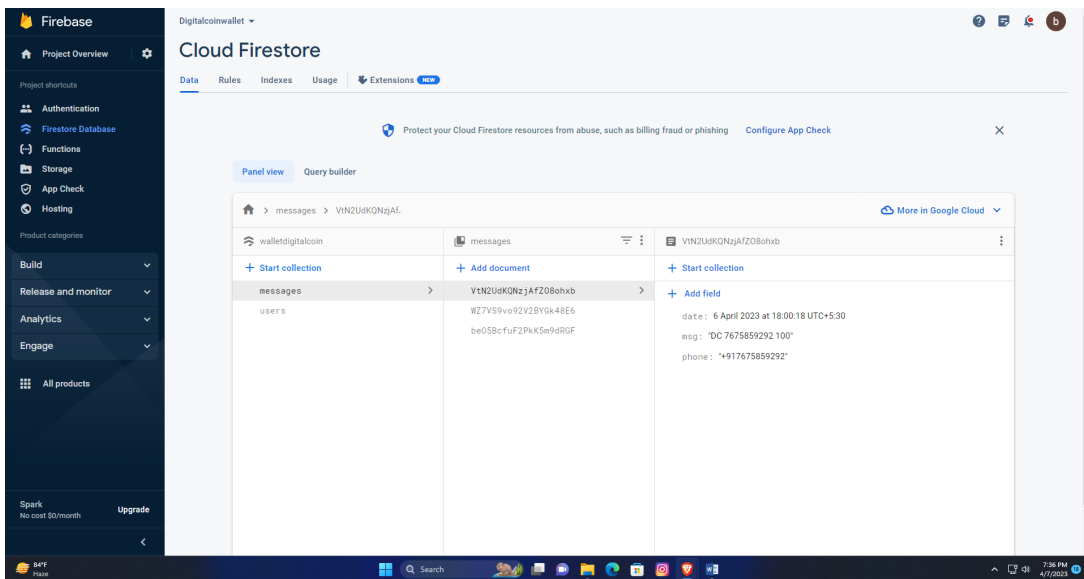


Fig 7.8: Firestore Database

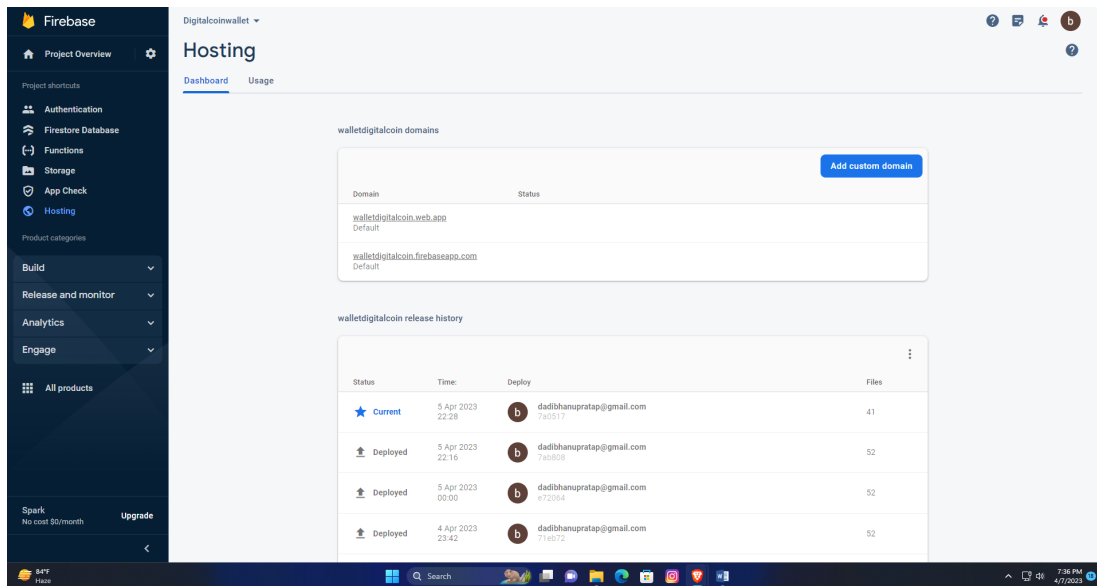


Fig 7.9: Hosting

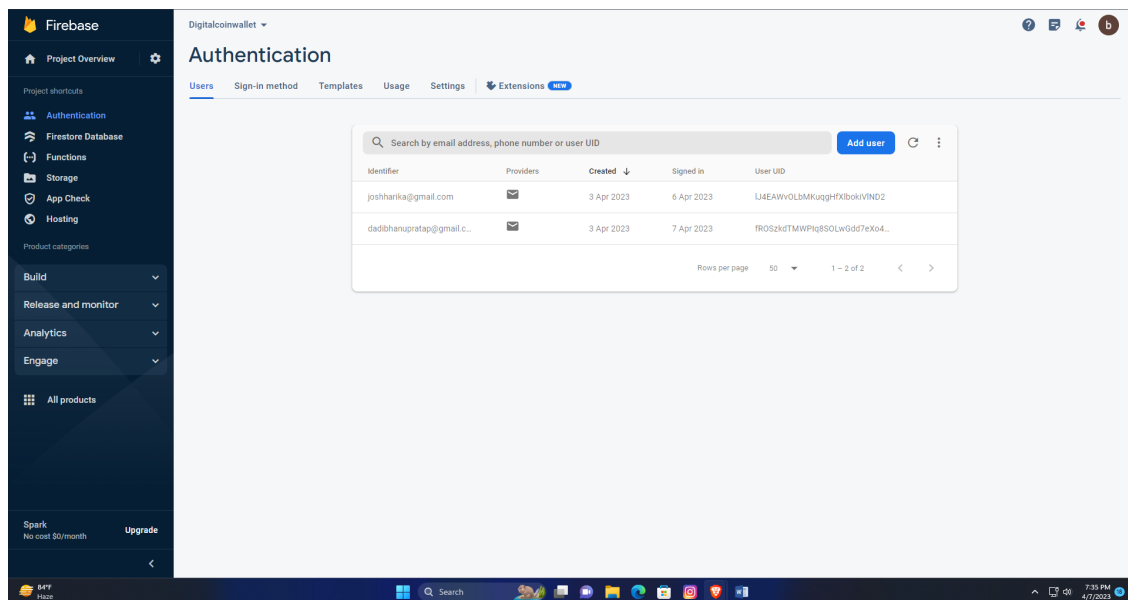


Fig 7.10: Authentication

8.SYSTEM TESTING

8.1.1 SOFTWARE TESTING

Software testing can be stated as the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by its design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy and usability. It mainly aims at measuring specification, functionality and performance of a software program or application.

Software testing can be divided into two steps:

1. **Verification:** it refers to the set of tasks that ensure that software correctly implements a specific function.
2. **Validation:** it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Software Testing can be broadly classified into two types:

Manual Testing:

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing. Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

Automation Testing:

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed

manually, quickly, and repeatedly. Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

8.1.2 TESTING METHODOLOGIES

Black box Testing: is the testing process in which a tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.

White box Testing: is the testing process in which a tester can perform testing on an application with having internal structural knowledge. Usually The Developers are involved in white box testing.

Gray Box Testing: is the process in which the combination of black box and white box techniques are used.

9. CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, digital wallets have become increasingly popular in recent years as a convenient and secure way to make transactions. They offer a range of benefits over traditional payment methods, including faster transaction times, lower fees, and improved security. The adoption of digital wallets is expected to continue to increase in the future, as more people become comfortable with the technology and more businesses begin to accept digital payments.

However, digital wallets also face challenges, such as concerns over security and privacy, and the need for standardization across different platforms. It is important for digital wallet providers to address these challenges in order to ensure continued growth and adoption of this technology.

Future Enhancements:

There are several areas where digital wallets could be improved in the future. One area is interoperability, which would allow users to transfer funds between different digital wallets and payment systems. This would increase convenience and reduce the need for users to maintain multiple accounts with different providers.

Another area for enhancement is the integration of additional features, such as loyalty programs, coupons, and other incentives to encourage users to make more transactions through digital wallets. This would increase the value proposition of digital wallets and help drive adoption.

Finally, digital wallets could benefit from further advances in biometric authentication technology, which would enhance security and reduce the risk of fraud. As biometric technology becomes more advanced and widely adopted, it is likely to become a standard feature of digital wallets.

10. REFERENCES

1. Chen, C. (2019). The Development and Prospect of Digital Wallets. *Journal of Business Research*, 98, 210-217.
2. Chiu, C. C., Lee, C. H., & Tan, K. H. (2020). A Study on Digital Wallets Adoption: A Focus on Convenience, Trust, and Innovativeness. *Telematics and Informatics*, 48, 101362.
3. Gharakhani, D., & Elyasi, M. (2019). Factors Influencing the Adoption of Digital Wallets: A Study of Iranian Consumers. *International Journal of Bank Marketing*, 37(1), 235-255.
4. Kim, J., & Kim, J. H. (2020). The Impact of Digital Wallets on Customers' Trust and Perceived Value. *Sustainability*, 12(14), 5731.
5. Lee, Y., & Chung, N. (2020). Factors Affecting Digital Wallet Adoption in Korea. *Journal of Retailing and Consumer Services*, 57, 102161.
6. Liao, C., Chen, Y., & Chen, H. (2019). The Relationship Between Trust and Continuance Intention of Mobile Payment Services: An Empirical Study of WeChat Wallet in China. *International Journal of Information Management*, 47, 219-227.
7. Lin, H. F., Wang, Y. S., & Luarn, P. (2018). Predicting Consumer Intentions to Adopt Mobile Wallets: Cross-Country Comparison between Taiwan and China. *Information Systems and e-Business Management*, 16(1), 81-104.

11. BIBLIOGRAPHY

1. Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Ethereum Whitepaper.
- Johnson, B. (2017). Digital Wallets: Trends, Opportunities, and Market Forecast. Juniper Research.
- Kshetri, N., & Voas, J. (2016). Blockchain-enabled digital government. *IT Professional*, 18(3), 11-14.
2. Li, X., Jiang, P., Chen, T., Luo, X., & Wen, Q. (2017). A survey on the security of blockchain systems. *Future Generation Computer Systems*, 82, 1-19.
3. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Bitcoin.org.
4. Tapscott, D., & Tapscott, A. (2016). Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world. Penguin.
5. Vigna, P., & Casey, M. J. (2015). The age of cryptocurrency: how bitcoin and digital money are challenging the global economic order. St. Martin's Press.
6. Wang, H., Xu, Z., & Xu, B. (2017). Group signature scheme based on blockchain in mobile payment. *Journal of Ambient Intelligence and Humanized Computing*, 8(6), 915-922.
7. Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology?—a systematic review. *PloS one*, 11(10), e0163477.