FULL STACK STUDENT INTERNSHIP AT VERIZON

DSN4096-CAPSTONE PROJECT PHASE-II

Submitted by

ISHITA VERMA
(20BCE10395)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

MAY 2024

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE
MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"FULL STACK STUDENT INTERNSHIP AT VERIZON"** is the bonafide work of "**ISHITA VERMA (20BCE10395)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR	**PROJECT GUIDE**
Dr J. Manikandan,,	Dr. Gaurav Soni
Senior Assistant Professor (Gr-2)	Senior Assistant Professor (Gr-2)
School of Computing Science and Engineering	School of Computing Science and Engineering
VIT BHOPAL UNIVERSITY	VIT BHOPAL UNIVERSITY

The Internship Examination is held on _____

OFFER LETTER

verizon✓

Verizon Data Services India Pvt. Ltd.
Regd. & Corp. Office:
8th Floor, Citius Block - B, Olympia Technology Park
Plot No. 1, SIDCO Industrial Estate
Guindy, CHENNAI - 600 032. INDIA

Tel. (91 44) 4394 5000
Fax. (91 44) 4394 4000
www.verizon.com

CIN: U72300TN2001PTC046551

INTERNSHIP AGREEMENT
(Private & Confidential)

Dear Ishita Verma,

We are pleased to offer you an internship at Verizon Data Services India Pvt. Ltd. (**VERIZON INDIA**) on the terms set out in this agreement (**Agreement**).

1. Internship

1.1 The content of your internship program will be determined by an assigned member from Human Resources, who will also be your main point of contact for further guidance during your internship program. The purpose of this internship program is to provide you with relevant guidance and to equip you with experience that will be helpful in your future career. In furtherance of this, you may be asked to execute various assignments/tasks from time to time during the course of your internship. You should use your best efforts in performing and delivering the tasks. During the tenure of your internship, you should work closely with your mentor, who will guide you and review your assignment to ensure that the internship is an enriching experience for you.

1.2 You will be on the physical premises of VERIZON INDIA and you are expected to comply with all applicable rules and regulations of VERIZON INDIA as enforced from time to time in respect of the matters not covered by this Agreement. VERIZON INDIA's decision on all such matters will be final and binding on you.

1.3 It is clarified that you are not an "employee" or a "workman" for the purposes of any employment statute.

1.4 Your internship is subject to satisfactory completion of background and reference checks, the absence of any bars or restrictions on your internship with VERIZON INDIA and your acceptance of these terms and conditions.

2. Location

2.1 You will be based at VERIZON INDIA office in 201 TITUS,. BLDG#10, Hyderabad. However, if required and at VERIZON INDIA's request, you may have to travel to certain parts of India / other Verizon India offices in your capacity as an intern.

3. Stipend and Certificate

3.1 You will be paid a stipend of INR 30,000 (Rupees Thirty Thousand Only) per month, payable once a month / on completion of the internship program, subject to applicable deductions. VERIZON INDIA may, subject to any relevant legal requirements, deduct from your stipend any amounts that you owe to VERIZON INDIA or as required under law. It is clarified that you will not be entitled to any other allowances or benefits given to the regular employees of VERIZON INDIA. The stipend paid to you for your internship with VERIZON INDIA is not and shall not be construed as "wages" for the purposes of any employment statute.

3.2 On your successful completion of the internship program, you will be provided an internship certificate by VERIZON INDIA.

Ishita Verma

verizon✓

4. Term of the Internship

4.1 Your internship program shall be for a term of 6 (six) months, starting on February 19, 2024 up until June 28, 2024.

5. Working Hours and Leave

5.1 For us to be able to effectively equip you with the skills you would require in your future professional career, you would be required to be available for 8 hours each day during the course of your internship.

5.2 During the course of your internship, in case you require to take leave for any personal emergencies, you would be required to take approval from your HRBP / Program POC prior to taking such leave.

6. Termination

6.1 During your internship program, either party may terminate this Agreement by giving 2 weeks' notice in writing.

6.2 VERIZON INDIA shall be entitled to terminate this Agreement with immediate effect (but without prejudice to the rights and remedies of VERIZON INDIA for any breach of this Agreement and to your continuing obligations under this Agreement) if you are guilty of dishonesty or serious or persistent misconduct, or without reasonable cause neglect or refuse to attend your internship or fail to perform any of your obligations hereunder or if you are found lacking in dedication towards your obligations hereunder, or fail to observe VERIZON INDIA's disciplinary rules or any other applicable regulations of VERIZON INDIA.

7. Limited Authority

7.1 You will not enter into any commitments or dealings on behalf of VERIZON INDIA or be a party to any alteration of any principle or policy of VERIZON INDIA or exceed the authority or direction vested in you without the previous sanction of VERIZON INDIA or those in authority over you. Any violation of this may lead to termination of your internship program with immediate effect.

8. Non-competition and Declaration of interest

8.1 You acknowledge that during the course of your internship with VERIZON INDIA, you may become familiar with VERIZON INDIA's trade secrets and the other confidential information concerning VERIZON INDIA, its associates, related companies and that your work may be of a special, unique and extraordinary value to VERIZON INDIA. You agree that during the term hereof, you shall not directly or indirectly own, manage, control, participate in, consult with, render services for (as an intern or otherwise), or engage in any business competing with VERIZON INDIA.

8.2 Without prejudice to the provisions of Clause 8.1 of this Agreement, you will notify VERIZON INDIA in writing of all business interests that you have directly or indirectly related to the business or activities of VERIZON INDIA.

9. Confidentiality and Intellectual Property

9.1 You will be required, as a condition of your internship with VERIZON INDIA, to sign the accompanying Business and Scientific Information and Security Agreement and the Confidentiality and Non-Solicitation Agreement, both of which are incorporated into this Agreement as Schedule 1 and Schedule 2 and form part of the terms of your internship.

10. Return of VERIZON INDIA Property

10.1 You will be responsible for the safe return of all the properties of VERIZON INDIA including but not limited to any electronic devices, Security Badge, drawings, software, employee data, notebooks, manuals, documents, computerization of technical data,

Sujata Verma

verizon✓

customer lists, specifications, files, memoranda, or other records of any nature belonging to VERIZON INDIA or any reproduction thereof which may have been provided to you during the course of your internship with VERIZON INDIA or which may be in your use, custody, care or charge. For the loss of any property of VERIZON INDIA in your possession, VERIZON INDIA will have a right to assess the value of the loss on its own basis and recover the damages as it deems proper in the event of your failure to account for such material or property to its satisfaction.

10.2 On expiry or earlier termination of this Agreement, you shall immediately hand over all papers, documents and other property of the VERIZON INDIA as may be in your possession, custody, control or power, including but not limited to any phones, computers, vehicles, etc. provided by VERIZON INDIA.

11. Data Privacy

11.1 As a part of your background check, as well as during the course of your internship with VERIZON INDIA, VERIZON INDIA may collect personal information, including but not limited to, your bank details, credit history, medical records and history, and such other personal data or information as it may deem necessary, from time to time, as necessary for the purposes of your internship. You hereby acknowledge and grant your consent to VERIZON INDIA collecting, using, processing, storing, disposing off, and transferring, whether to any other VERIZON INDIA affiliate company, or to third party service providers, within or outside India, any such personal information relating to you.

12. Tax Liability

12.1 You will arrange to take care of your tax liabilities, i.e. income tax or any other tax as may be applicable to the aforesaid stipend payments, and VERIZON INDIA will not be liable for the same. All stipend and other payments (if any) to you will be subject to tax withholding in accordance with applicable laws.

13. Governing Law and Dispute Resolution

13.1 This Agreement shall be governed by and construed in accordance with the laws of India. The courts in Chennai will have exclusive jurisdiction in relation to all disputes arising out of this Agreement.

This Agreement constitutes the entire terms and conditions governing your engagement as an intern with VERIZON INDIA.

If the terms and conditions of this agreement are understood and accepted, please sign and return the attached copy of this Agreement.

Yours sincerely,
For Verizon Data Services India Pvt. Ltd.

Samir Singh
Director - Talent Acquisition

Acknowledgement

I hereby accept the internship assignment on the terms and conditions set out in the foregoing Agreement.

verizon

Signature: Ishita Verma
Name: Ishita Verma
Date: 14 Feb, 2024

Regd. & Corporate Office: 8th Floor, Citius Block- B, Olympia Technology Park. Plot no. 1, SIDCO Industrial Estate, Guindy. Chennai-600 032 . India.

Signature: Ishita Verma
Email: ishita.verma064@gmail.com

ACKNOWLEDGEMENT

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I would like to express my sincere gratitude to Verizon for offering me the opportunity to work as an SDE intern and my supervisor Mr. Venkata Avinash Sainni for his esteemed guidance and actively participating in my project.

I would like to thank my guide Dr. Gaurav Soni, for continually guiding and giving valuable suggestions to complete the project work.

I wish to express my heartfelt gratitude to Dr. J. Manikandan, Head of the Department, School of Computing Science and Engineering for much of his valuable support and encouragement in carrying out this work.

I would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

1.	SDE	Software Development Engineer
2.	VZ	Verizon
3.	CSS	Cascading Style Sheet
4.	DB	Database
5.	UI	User Interface
6.	UX	User Experience
7.	CRUD	Create, Read, Update, Delete
8.	CI/CD	Continuous Integration/Continuous Deployment
9.	API	Application Programming Interface
10.	JS	JavaScript
11.	REST	Representational State Transfer
12.	JWT	JSON Web Token
13.	SQL	Structured Query Language
14.	JSON	JavaScript Object Notation

LIST OF FIGURES AND GRAPHS

ABSTRACT

Embarking on an enriching journey as a Software Development Engineering (SDE) intern at Verizon (VZ), I have had the privilege of immersing myself in a stimulating environment of learning and professional development. The internship commenced with a structured curriculum encompassing foundational and technical training sessions, meticulously designed to equip interns with the requisite knowledge and skills to excel in their roles.

Following the initial training phase, interns were grouped into teams and assigned diverse projects aligned with Verizon's technological ecosystem. These projects provided a hands-on opportunity to apply theoretical concepts to real-world scenarios, fostering collaboration, innovation, and problem-solving within a supportive team environment.

As an SDE intern, I have actively engaged in project development, leveraging my technical acumen to contribute meaningfully to the team's objectives. Through continuous iteration and feedback cycles, I have honed my coding abilities, expanded my proficiency in various technology stacks, and cultivated a deeper understanding of software development best practices.

Moreover, the internship experience extends beyond technical aspects, encompassing invaluable opportunities for personal and professional growth.

Engaging in cross-functional collaboration, participating in mentorship programs, and attending networking events have facilitated the development of essential soft skills such as communication, teamwork, and adaptability.

As the internship progresses, the transition into functional training further enriches the learning experience, providing insights into Verizon's operational framework and strategic objectives. This comprehensive approach to training underscores Verizon's commitment to nurturing talent and fostering a culture of continuous learning and development.

In this report, I reflect on the myriad experiences, challenges, and triumphs encountered during my internship at Verizon. Through a structured narrative, I delve into the projects undertaken, technical competencies acquired, and the broader lessons gleaned from this immersive experience. As I continue to immerse myself in the vibrant ecosystem of Verizon, I am grateful for the opportunities bestowed upon me and eagerly anticipate the contributions I can make to the organization and beyond.

TABLE OF CONTENTS

CHAPTER-1:
PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

In the contemporary landscape of digital innovation, organizations strive to optimize their customer experiences by leveraging advanced technologies and robust infrastructures. This report encapsulates the journey undertaken during my internship at Verizon, focusing on the development and enhancement of the BankWeb project. This project served as a pivotal point in acquiring hands-on experience in full-stack development and understanding the intricate dynamics of customer-centric solutions.

1.2 Motivation for the work

The motivation for undertaking this project stems from the increasing demand for seamless digital experiences and the need to stay competitive in today's dynamic market. Recognizing the significance of delivering efficient and user-centric solutions, Verizon works on enhancing its platform's capabilities and meet evolving customer expectations. The impetus behind the BankWeb project stemmed from the ever-growing demand for seamless, user-friendly digital banking solutions. With the proliferation of online transactions and the shift towards digital platforms, there arose a need for a comprehensive banking web application that could cater to diverse user requirements.

1.3 Company Overview

Verizon is one of the largest telecommunications companies globally, renowned for its extensive network infrastructure and innovative solutions in mobile and broadband services. With a strong presence in various segments of the telecommunications industry, Verizon is committed to delivering reliable connectivity and cutting-edge technologies to consumers and businesses alike. Verizon's mobile network is the second-largest wireless carrier in the United States, with 144.8 million subscribers as of December 31, 2023.

Fig-1

Visible, a subsidiary of Verizon, is a digital wireless carrier offering affordable and simplified mobile phone services. Launched in 2018, Visible operates on Verizon's extensive network, providing customers with reliable coverage and high-speed data. Visible distinguishes itself by offering straightforward pricing, and a seamless digital experience.

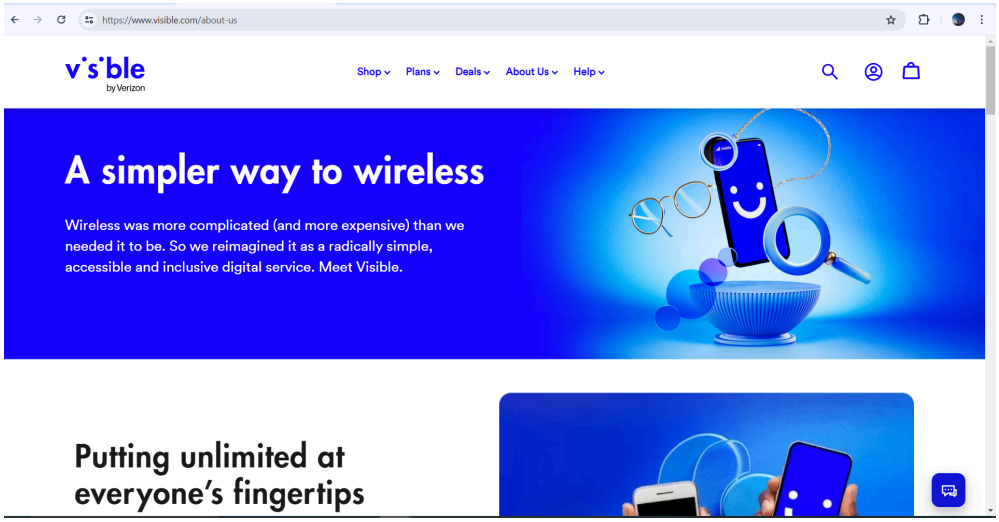

Fig-2

As part of Verizon's strategic initiatives, Visible aims to disrupt the traditional wireless carrier model by focusing on simplicity, transparency, and customer-centric services. With its innovative approach and competitive offerings, Visible continues to attract consumers seeking hassle-free mobile solutions without compromising on network quality or performance.

1.4 **Problem Statement**

Within the scope of the internship program at Verizon, the challenge was to bridge the gap between theoretical knowledge gained during foundational and technical training and its practical application in real-world projects. The objective was twofold: firstly, to develop a comprehensive understanding of software development methodologies, tools, and technologies through hands-on experience, and secondly, to contribute meaningfully to the enhancement of Visible, a prominent product within the Verizon ecosystem.

The BankWeb project, undertaken as part of an internship program at Verizon, aimed to address several key challenges in the banking sector related to user experience, transaction security, and accessibility. The objective was to develop a robust, user-friendly, and scalable banking application that meets the evolving needs of both customers and stakeholders, while also providing a learning platform for interns to gain practical experience in full-stack development and project management.

1.5 **Objective of the work**

- Technical Mastery: The primary objective of the internship program is to provide hands-on experience in applying their technical knowledge and skills to real-world projects.
- Contribution to Visible Enhancement: Identifying opportunities for improvement and implementing innovative solutions. This involves understanding the platform's architecture, functionalities, and user base, and leveraging acquired skills to propose and implement enhancements that enhance user experience and platform performance.
- Skill Development and Growth: The internship program aims to facilitate interns' professional growth and development by providing opportunities for learning, mentorship, and skill enhancement.
- Feedback and Reflection: This feedback-driven approach helps to iterate on their work, learn from their mistakes, and grow as professionals.

1.6 **Organization of the project**

The project organization is structured to facilitate seamless collaboration and effective execution of tasks within the internship program. It encompasses the following key aspects:

- Team Structure: Interns are organized into cross-functional teams, each comprising individuals with diverse skill sets and backgrounds. This facilitates knowledge sharing, peer learning, and the distribution of responsibilities based on individual strengths and expertise.
- Project Management: Project management methodologies, such as Agile or Scrum, are employed to streamline project workflows, prioritize tasks, and ensure timely delivery of milestones.
- Task Allocation: This ensures that each team member is engaged in activities that align with their development objectives while contributing meaningfully to the project's overall success.

- Communication Channels: Open communication channels, including team meetings, instant messaging platforms, to facilitate real-time collaboration and information sharing among team members.
- Documentation and Reporting: Comprehensive documentation of project activities, including progress reports,contribution from members and technical documentation, is maintained to track project milestones, document decisions, and facilitate knowledge transfer among team members.

1.7 **Summary**

In summary, the project aims to leverage interns' technical expertise and problem-solving skills to enhance the user experience, and overall performance.The project's organization emphasizes the importance of clear communication, collaboration, and alignment with organizational objectives to deliver impactful outcomes. By successfully navigating through the various project phases, interns not only gain valuable hands-on experience but also contribute to the ongoing evolution.

CHAPTER-2:
RELATED WORK INVESTIGATION

2.1 **Introduction**

The related work investigation serves as a foundational step in understanding the landscape of existing solutions and methodologies relevant to both the Visible platform and my intern project, BankWeb. This chapter aims to explore the core areas of these projects, analyze existing approaches and methods, evaluate their strengths and weaknesses, identify key issues or observations, and provide a summary of findings to guide the subsequent phases of the projects.

2.2 **Core area of the project**

For the Visible platform, the core area of the project revolves around telecommunications services, customer experience management, and platform scalability. Understanding how Visible operates within the broader Verizon ecosystem, its target market, and its unique value proposition is essential for contextualizing the related work investigation.

In the case of my intern project, BankWeb, the core area focuses on online banking services, user authentication, transaction processing, and customer interaction. Examining existing online banking platforms, security protocols, and user interface designs will be crucial for informing the development of BankWeb.

2.3 **Existing Approaches/Methods**

For the Visible platform, existing approaches and methods may include:

- Mobile app development frameworks such as React Native or Flutter.
- API management solutions like APIGEE for handling backend services and integrations.
- Customer experience optimization techniques such as user interface design principles and performance optimization strategies.

In the case of the BankWeb project, existing approaches and methods may involve:

- Web development frameworks like React.js for building interactive user interfaces.
- Backend technologies such as Spring Boot for developing RESTful APIs and microservices.

- Security protocols and practices for ensuring data confidentiality and integrity in online banking systems.

2.4 **Pros and cons of the stated Approaches/Methods**

Pros and cons of the existing approaches/methods vary depending on factors such as scalability, performance, security, and user experience. For example:

- React Native offers cross-platform compatibility and faster development cycles but may face limitations in accessing device-specific features compared to native development.
- Spring Boot provides robust support for building microservices and simplifies configuration but may introduce overhead in terms of memory consumption and startup time.

2.5 **Issues/observations from investigation**

During the investigation, several issues or observations may arise, such as:

- Compatibility challenges between different versions of libraries or frameworks.
- Performance bottlenecks in API responses or frontend rendering.
- Security vulnerabilities in authentication or data transmission processes.

2.6 **Summary**

In summary, the related work investigation sheds light on the core areas of the Visible platform and the BankWeb project, explores existing approaches and methods, evaluates their pros and cons, and identifies potential issues or observations. This analysis serves as a foundation for informing decision-making processes and guiding the development efforts in subsequent chapters.

CHAPTER-3:
REQUIREMENT ARTIFACTS

3.1 **Introduction**

This chapter delves into the specific project requirements essential for the development and implementation of my project, online banking web application. It encompasses both the requirements pertinent to my project and those essential for integration with the Visible ecosystem.

3.2 **Hardware and Software requirements**

Hardware Requirements:

- Desktop or laptop computer capable of running modern web browsers.
- Minimum: Dual-core processor (e.g., Intel Core i3 or AMD equivalent)
- Recommended: Quad-core processor or higher
- Memory (RAM): Minimum: 4 GB RAM
- Storage: Minimum: 128 GB solid-state drive (SSD) or hard disk drive (HDD)
- Stable broadband internet connection for accessing the online banking application without interruptions

Software Requirements:

- Operating System:
    - Windows 10, macOS, or Linux (Ubuntu, Fedora, etc.)
    - Web Browser such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge
    - Ensure compatibility with modern web standards and security protocols
    - Integrated Development Environment (IDE) such as Visual Studio Code, IntelliJ IDEA, or Eclipse for developers working on the project
    - Database Management System (DBMS) for storing user data, transaction records, and other relevant information. Examples include MySQL, PostgreSQL, or Microsoft SQL Server

- Backend Framework:
    - Java-based Spring Boot framework for building microservices and RESTful APIs

    - Node.js with Express.js or Python with Django are also viable options depending on the project requirements
- Frontend Framework:
    - React.js for building dynamic and responsive user interfaces
- Git for version control and collaboration among team members
    - Platforms like GitHub or GitLab for hosting repositories and managing project workflows
- API Management Platform:
    - Apigee or similar API management platforms for securing, managing, and analyzing APIs used in the application

3.3 **Specific Project requirements**

For development of my assigned intern project, BankWeb, the tools and frameworks required are:

- ReactJs, CSS/Tailwind CSS for front-end development.
- Spring Boot for backend microservices.
- APIGEE for API management.
- Java programming language for backend development.
- SQL database/H2 in-memory database for data storage.

3.3.1 Data Requirement

- Secure storage and management of user account information, transaction records, and other sensitive data.
- Integration with existing banking systems or APIs to retrieve and update account information in real-time.

3.3.2 Functions Requirement

- User authentication and authorization mechanisms to ensure secure access to account information and transactions.
- Ability for users to initiate fund transfers, payments, and other banking transactions seamlessly.
- Provision of account management functionalities such as balance inquiries, transaction history viewing, and account profile updates.

3.3.3 Performance and Security Requirement

- High availability and reliability of the application to ensure uninterrupted banking services for users.
- Implementation of robust security measures including encryption, authentication, and authorization to safeguard user data and transactions.
- Compliance with industry standards and regulations.

3.3.4 Look and Feel Requirements

- Intuitive and user-friendly interface design to enhance the user experience and ease of navigation.
- Consistent branding and visual elements aligned with the bank's corporate identity and brand guidelines.

3.4 **Summary**

This chapter summarizes the hardware and software requirements as well as the specific project requirements necessary for the successful development and implementation of the BankWeb online banking web application. These requirements serve as the foundation for designing and building a robust and user-centric banking platform.

CHAPTER-4:
DESIGN METHODOLOGY AND ITS NOVELTY

4.1 **Methodology and goal**

The design methodology adopted for the development of BankWeb follows a structured approach aimed at achieving the project goals effectively and efficiently. The methodology encompasses various stages such as requirements analysis, system design, implementation, testing, and deployment, ensuring a systematic and organized development process.

4.2 **Functional modules design and analysis**

- User Authentication Module: Implements authentication mechanisms to verify user identity securely.
- Transaction Management Module: Facilitates fund transfers, payments, and transaction processing.
- Account Management Module: Allows users to view account details, transaction history, and manage their profiles.
- Testimonials Module: Enables users to post reviews and testimonials about their banking experience.
- FAQs Module: Provides a repository of frequently asked questions and answers for user assistance.
- Cards Module: Allows users to manage their bank cards and apply for new cards if needed.

4.3 **Software Architectural designs**

The software architecture of BankWeb follows a microservices-based architecture to ensure modularity, scalability, and maintainability. It comprises the following key architectural components:

- Developed using React framework for building responsive and interactive user interfaces.
- Implemented as Spring Boot-based microservices to handle business logic, data processing, and API integrations.
- Utilizes H2 database for storing and retrieving user data, account information, and transaction records.

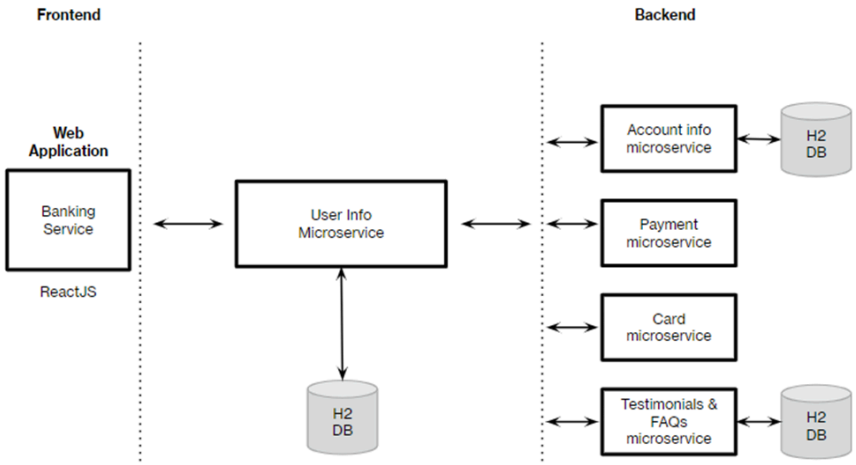

Fig-3

4.4 **Subsystem services**

BankWeb incorporates various subsystem services to support its functionalities, including:

- User authentication and authorization services.
- Transaction processing and management services.
- Account management and profile services.
- Testimonials and FAQs management services.
- Card management and application services.

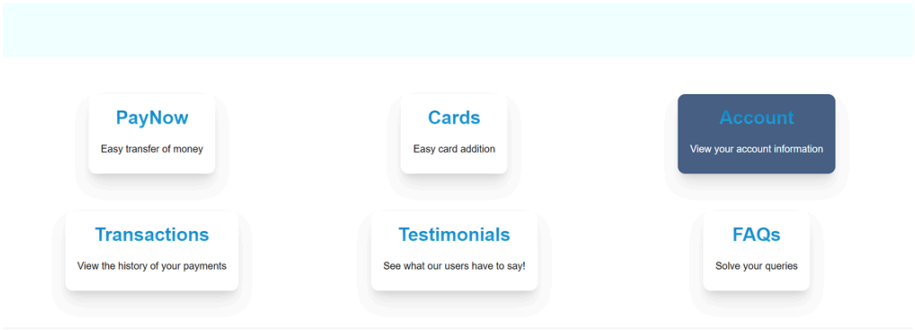

Fig-4

4.5 **User Interface designs**

The user interface designs of BankWeb are crafted to provide a seamless and intuitive banking experience.

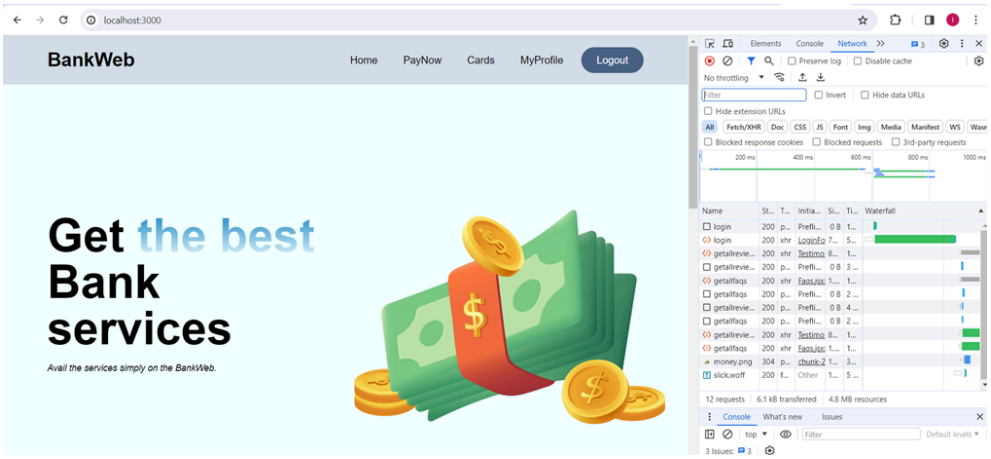
localhost:3000
BankWeb
Home
PayNow
Cards
MyProfile
Logout
Get the best
Bank
services
Avail the services simply on the BankWeb.
Elements
Console
Network
Preserve log
Disable cache
No throttling
Hide extension URLs
Blocked response cookies
Blocked requests
3rd-party requests
Name
Waterfall
12 requests
6.1 kB transferred
4.8 MB resources
Console
What's new
Issues

Fig-5

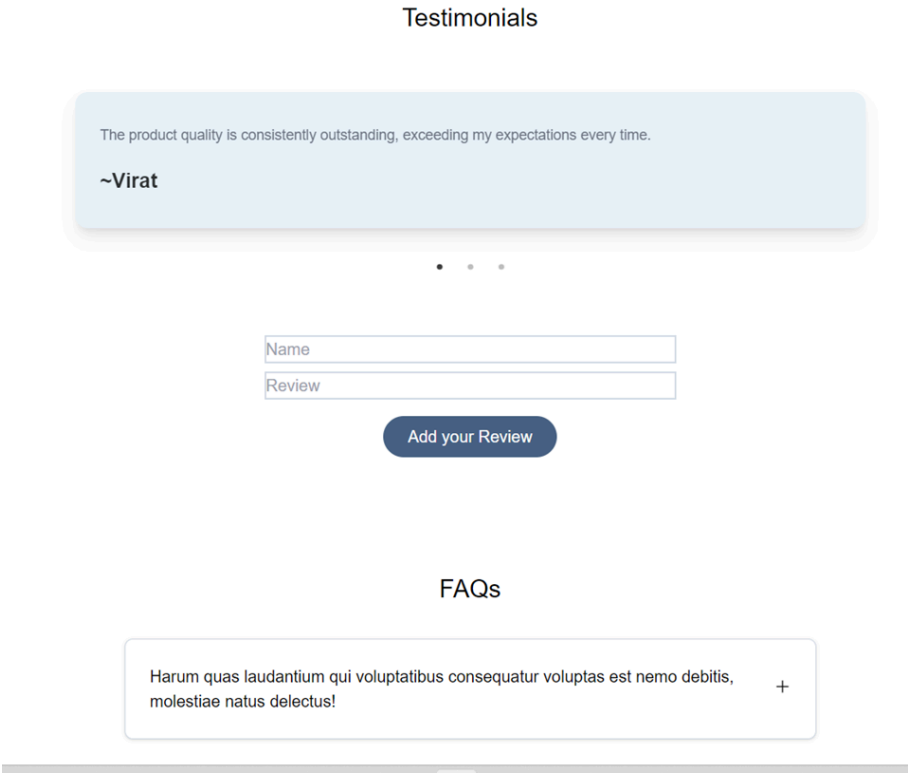
Testimonials
The product quality is consistently outstanding, exceeding my expectations every time.
~Virat
Name
Review
Add your Review
FAQs
Harum quas laudantium qui voluptatibus consequatur voluptas est nemo debitis, molestiae natus delectus!

Fig-6

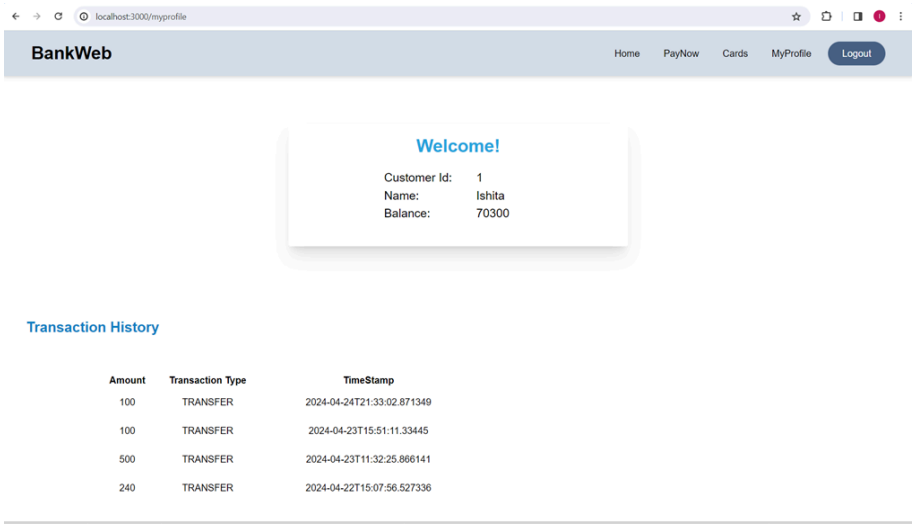

Fig-7

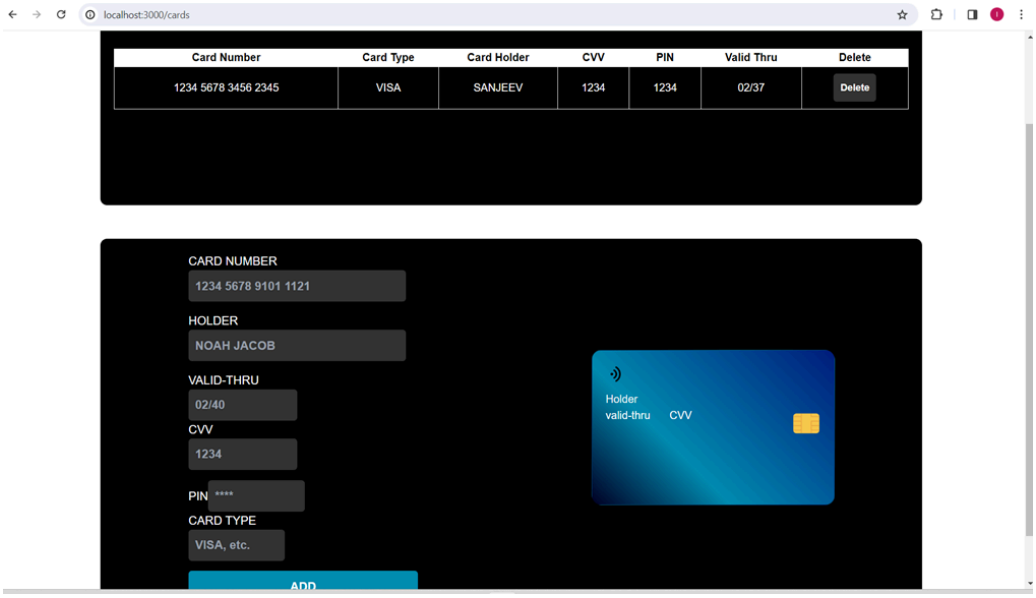

Fig-8

4.6 **Summary**

This chapter provides an overview of the design methodology adopted for the development of BankWeb, outlining the functional modules, software architectural designs, subsystem services, and user interface designs. The systematic approach ensures the successful implementation of the project goals and objectives, resulting in a robust and user-centric online banking web application.

CHAPTER-5:
TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 **Outline**

This chapter delves into the technical implementation of the BankWeb project, providing an overview of the development process and the key components involved. It outlines the various stages of implementation, including coding, form layout design, prototype submission, testing, validation, and performance analysis.

5.2 **Technical coding and code solutions**

Here are the implementation details of the backend and frontend components of BankWeb.

App,js:

The App component serves as the entry point of the BankWeb application. It defines the routing configuration using React Router, directing users to different pages based on the URL path.

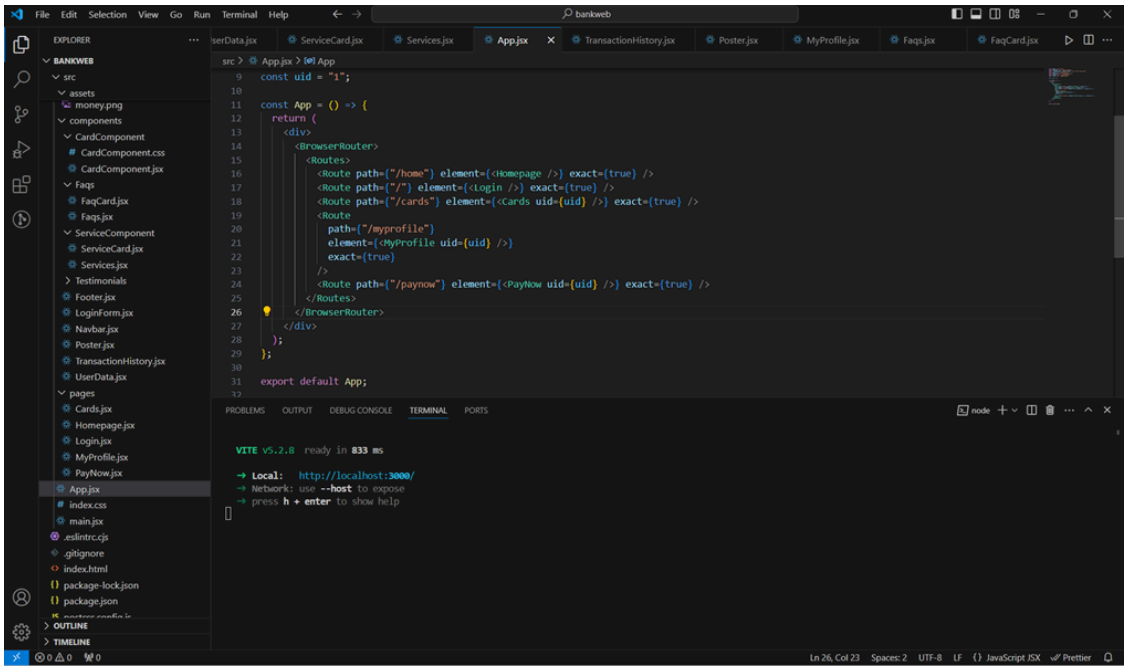

Fig-9

UserData.jsx:

UserData is a React component that fetches and displays user data from the server. It utilizes the useEffect hook to asynchronously retrieve user information upon component mounting, using the stored JWT token for authentication.

```
useEffect(() => {
  const fetchUserData = async () => {
  try {
     const token = localStorage.getItem("jwtToken");
            const response = await axios.get(
            "http://localhost:8080/api/current-user",
            {
            headers: {
            Authorization: `Bearer ${token}`,
            },
            }
            );
            setUserData(response.data);
  } catch (error) {
            // Handle error
            console.error("Failed to fetch user data:", error);
  }
  };

  fetchUserData();
 }, []);
```

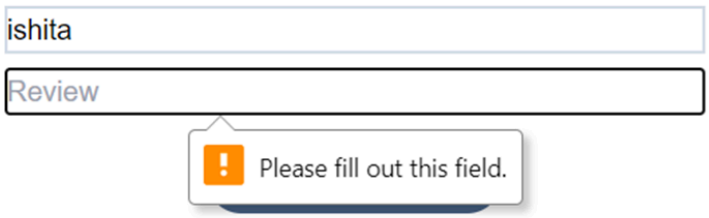

Fig-10

5.3 **Working Layout of Forms**

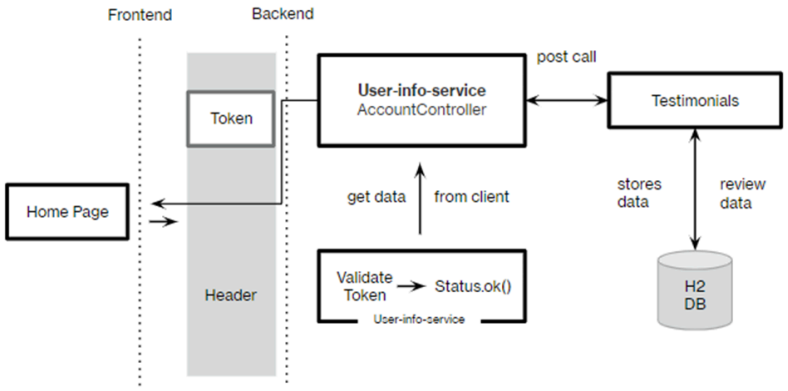

Fig-11

LoginForm.jsx:

This file contains the implementation of the login form component. It handles user authentication by sending a POST request to the server with the provided credentials. Upon successful login, it stores the JWT token in local storage and updates the login state.

```
const handleLogin = async () => {
  try {
  const response = await axios.post(
            "http://localhost:8080/api/auth/login",
```

```
            { username, password }
    );
  const { jwtToken } = response.data;
  localStorage.setItem("jwtToken", jwtToken);
  setLoggedIn(true); // Set logged in state to true
  } catch (error) {
  // Handle login error
  console.error("Login failed:", error);
  }
};
```

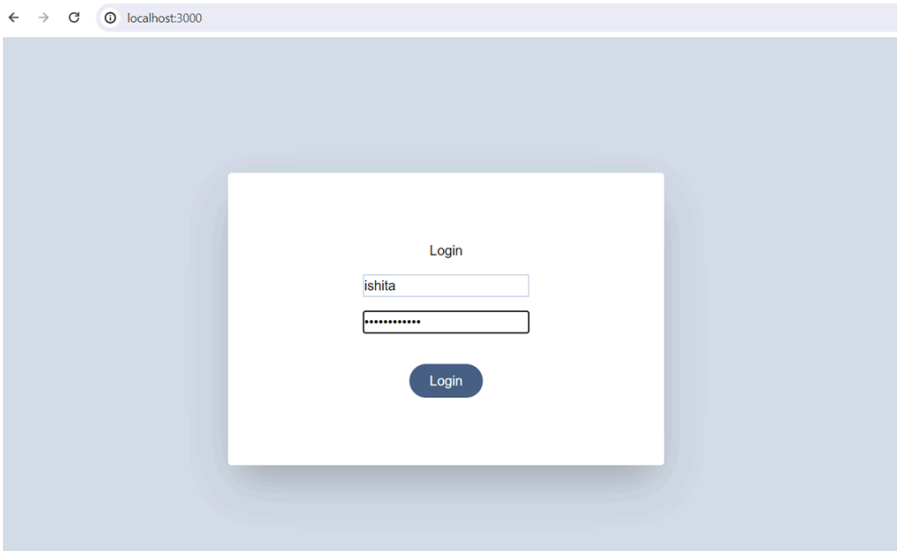

Fig-12

PayNow.jsx:

PayNow is a React component responsible for facilitating fund transfers between accounts. It renders a form where users can input recipient details and the transfer amount. Upon form submission, it sends a POST request to the server with the transfer details and handles the response accordingly.

```
const PayNow = ({ uid }) => {
 const [user, setUser] = useState({
  fromAccountId: uid,
  toAccountId: "",
  amount: "",
 });
 const [result, setResult] = useState("");


 const handleChange = (event) => {
  let value = event.target.value;
  let name = event.target.name;
  setUser((prevUser) => ({ ...prevUser, [name]: value }));
 };


 const handleSubmit = async (event) => {
  event.preventDefault();
  try {
  const token = localStorage.getItem("jwtToken");
  const response = await axios.post(
              "http://localhost:8080/api/accounts/transfer",
              user,
              {
              headers: {
              Authorization: `Bearer ${token}`,
              },
              }
```

```
  );
  setUser({
            fromAccountId: uid,
            toAccountId: "",
            amount: "",
  });
  setResult("Paid successfully!");
  console.log("Transaction done:", response.data);
  } catch (error) {
  console.log("Error:", error);
  }
};

return (
  <>
  <Navbar />
  <form
            onSubmit={handleSubmit}
            className="items-center flex flex-col gap-3 py-40"
  >
            <label className="flex flex-row gap-4">
            <p>Your Account id: </p>
            <input
            className="border-2 border-solid border-navbg min-w-[150px] md:w-[200px]"
            name="fromAccountId"
            value={user.fromAccountId}
```

```
placeholder="Transfer from account id"
disabled
/>
</label>

<input
className="border-2 border-solid border-navbg md:w-[400px] min-w-[300px]"
type="number"
step="1"
name="toAccountId"
value={user.toAccountId}
onChange={handleChange}
placeholder="Transfer to account id"
/>
<input
className="border-2 border-solid border-navbg md:w-[400px] min-w-[300px]"
type="number"
step="10.00"
name="amount"
value={user.amount}
onChange={handleChange}
placeholder="Amount (Eg: 100.00)"
/>
<p className="text-secondary">{result}</p>
<button
type="submit"
```

```
            className="bg-navyblue text-white rounded-full py-2 px-6 hover:bg-opacity-90"
            >
            Transfer
            </button>
  </form>
  </>
 );
};


export default PayNow;
```

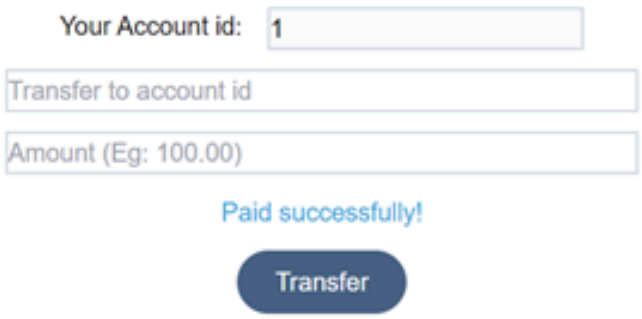

Fig-13

5.4 **Backend Services**

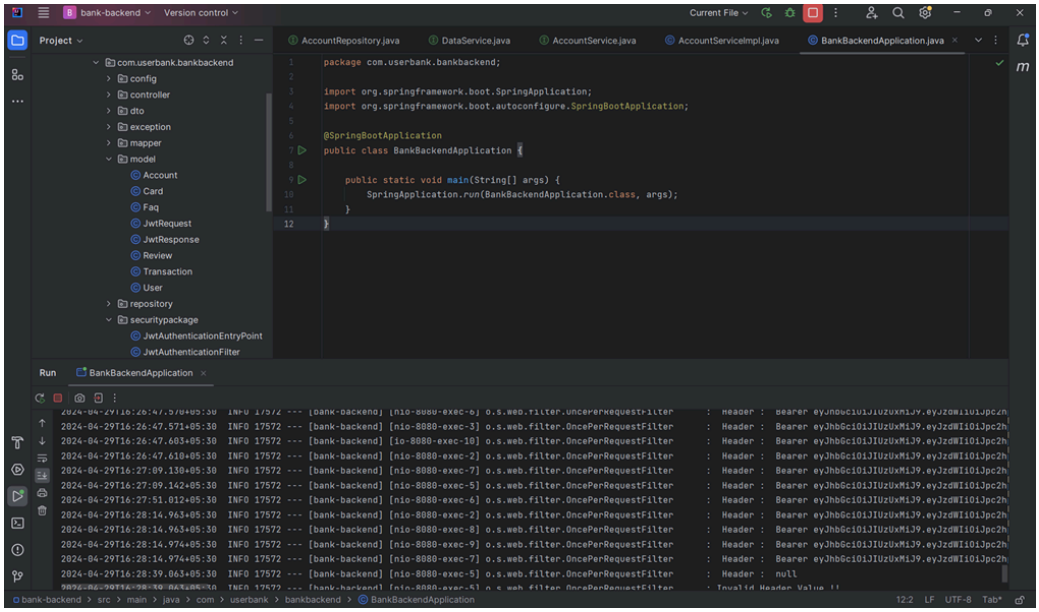

Fig-14

AccountController.java:

AccountController is a Spring Boot REST controller responsible for handling HTTP requests related to account management. It defines endpoints for account creation, retrieval, deposit, withdrawal, deletion, fund transfer, and transaction retrieval.

```
@CrossOrigin("*")
@RestController
@RequestMapping("/api/accounts")
public class AccountController {
  private AccountService accountService;

  public AccountController(AccountService accountService) {
  this.accountService = accountService;
  }

  @PostMapping
```

```
 public ResponseEntity<AccountDto> addAccount(@RequestBody AccountDto accountDto){

  return new ResponseEntity<>(accountService.createAccount(accountDto), HttpStatus.CREATED);

  }


  @GetMapping("/{id}")

  public ResponseEntity<AccountDto> getAccountById(@PathVariable Long id){

     AccountDto accountDto=accountService.getAccountById(id);

  return ResponseEntity.ok(accountDto);

  }

  // Delete Account REST API

  @DeleteMapping("/{id}")

  public ResponseEntity<String> deleteAccount(@PathVariable Long id){

   accountService.deleteAccount(id);

   return ResponseEntity.ok("Account is deleted successfully!");

  }

  @GetMapping("/{id}/transactions")

     public  ResponseEntity<List<TransactionDto>>  fetchAccountTransactions(@PathVariable("id")  Long
accountId){

     List<TransactionDto> transactions=accountService.getAccountTransactions(accountId);

   return ResponseEntity.ok(transactions);

  }


}
```

AuthController.java:

AuthController is a Spring Boot REST controller responsible for user authentication. It defines an endpoint for user login, which validates user credentials, generates a JWT token upon successful authentication, and returns the token to the client.

```java
@RestController
@RequestMapping("/api/auth")
public class AuthController {
  @Autowired
  private UserDetailsService userDetailsService;
   @Autowired
  private AuthenticationManager manager;
   @Autowired
  private JwtHelper helper;
  private Logger logger = LoggerFactory.getLogger(AuthController.class);

  @PostMapping("/login")
  public ResponseEntity<JwtResponse> login(@RequestBody JwtRequest request) {

  this.doAuthenticate(request.getUsername(), request.getPassword());
      UserDetails userDetails = userDetailsService.loadUserByUsername(request.getUsername());
     String token = this.helper.generateToken(userDetails);

     JwtResponse response = JwtResponse.builder()
             .jwtToken(token)
             .username(userDetails.getUsername()).build();
  return new ResponseEntity<>(response, HttpStatus.OK);
  }

  private void doAuthenticate(String email, String password) {
     UsernamePasswordAuthenticationToken authentication =
             new UsernamePasswordAuthenticationToken(email, password);
```

```
 try {
            manager.authenticate(authentication);
 } catch (BadCredentialsException e) {
            throw new BadCredentialsException(" Invalid Username or Password  !!");
 }
 }
 @ExceptionHandler(BadCredentialsException.class)
 public String exceptionHandler() {
 return "Credentials Invalid !!";
 }
}
```

H2 Database:

The H2 database is utilized as the backend database management system for the BankWeb application. It offers lightweight, fast, and embeddable features, making it suitable for development and testing purposes.

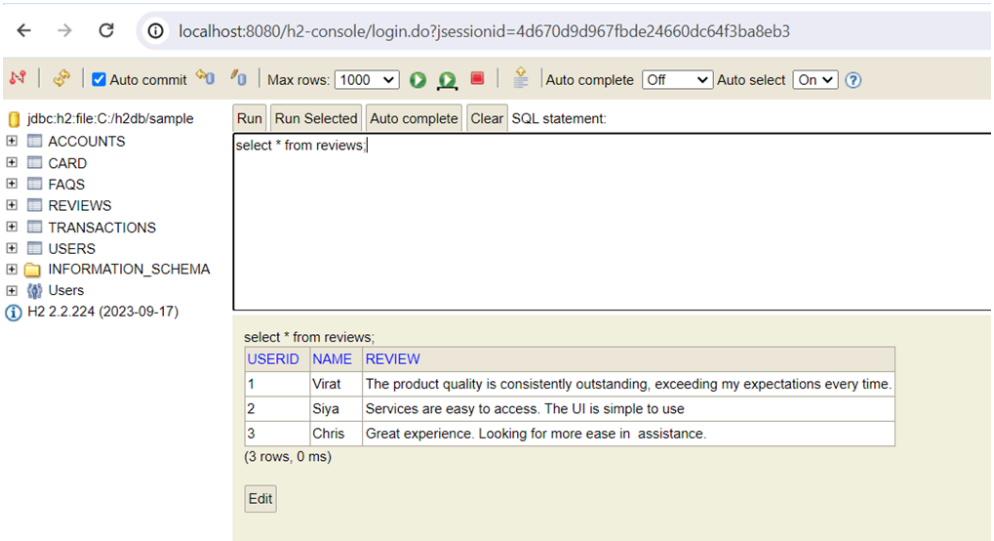

Fig-15

5.5 **Test and validation**

User Validation using JWT Tokens :

- JWT Token Generation
    - Login: Users authenticate by providing their credentials .
    - Token Generation: Upon successful authentication , the server generates a JWT token.
    - Token Storage: Users store the JWT token in local storage .
- Token Validation
    - Authorization Header : Users must include the token in the Authorization header of their API requests.
    - Token Expiry: Tokens have an expiration time to ensure security.

JwtAuthenticationEntryPoint.java:

JwtAuthenticationEntryPoint is a Spring component that handles unauthorized access attempts. It intercepts requests without valid JWT tokens and responds with an error message indicating access denial.

```
@Component

public class JwtAuthenticationEntryPoint implements AuthenticationEntryPoint {  @Override

        public void commence(HttpServletRequest request, HttpServletResponse response,
AuthenticationException authException) throws IOException, ServletException {

    response.setStatus(HttpServletResponse.SC_UNAUTHORIZED);

    PrintWriter writer = response.getWriter();

    writer.println("Access Denied !! " + authException.getMessage());

  }

}
```

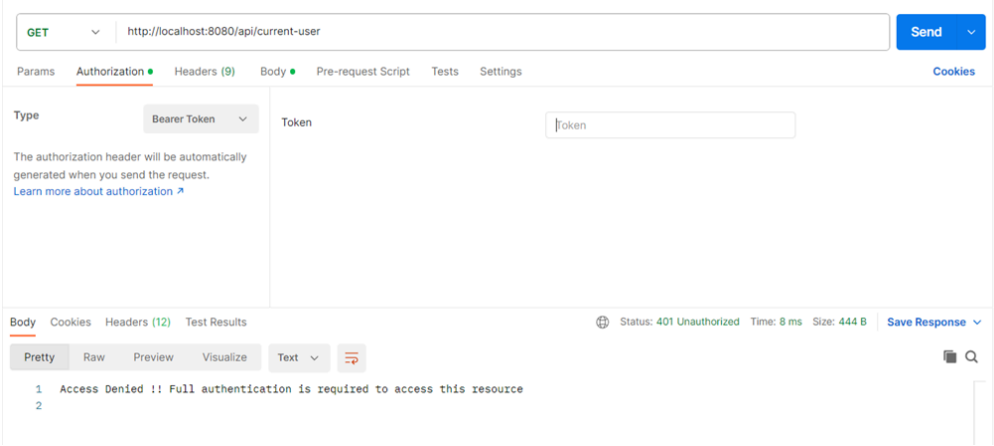

Fig-16

5.6 **Summary**

It outlines the overall structure of the application, including the frontend components built with React and the backend services implemented using Spring Boot. Additionally, it discusses the integration of the H2 database and the authentication mechanism using JWT tokens. The chapter concludes with an analysis of the application's performance and functionality, highlighting areas of improvement and future considerations.

CHAPTER-6:
PROJECT OUTCOME AND APPLICABILITY

6.1 **Outline**

This section provides an overview of the outcomes and applicability of the BankWeb project, highlighting key achievements and potential areas for further development.

6.2 **Key implementations outlines of the System**

The BankWeb project incorporates several key implementations to facilitate online banking functionalities. These include:

- User authentication and authorization mechanisms using JWT tokens for secure access.
- Integration of frontend components developed with React to create a dynamic and user-friendly interface.
- Backend services implemented with Spring Boot to handle business logic and database interactions.
- Utilization of the H2 database as the backend data management system for storing user account details and transaction history.

6.3 **Significant project outcomes**

The project has achieved the following significant outcomes:

- Successful implementation of core banking functionalities such as fund transfers, account management, and transaction history viewing.
- Seamless integration of frontend and backend components, ensuring a smooth user experience.
- Robust security measures implemented through JWT token-based authentication, safeguarding user data and transactions.
- Efficient handling of user interactions and data processing, resulting in a responsive and scalable application.

6.4 **Project applicability on Real-world applications**

The BankWeb project demonstrates applicability to real-world banking scenarios by offering a secure and user-friendly online banking platform. It can be utilized by financial institutions to provide their customers with convenient access to banking services from anywhere, at any time. The project's modular architecture and customizable features make it adaptable to various banking requirements and scalable for future enhancements.

6.6 **Inference**

In conclusion, the BankWeb project has successfully achieved its objectives of developing an online banking web application. It demonstrates the feasibility and effectiveness of using modern technologies such as React, Spring Boot, and JWT tokens to create secure and efficient banking solutions. Moving forward, continuous refinement and updates will be essential to meet evolving user needs and industry standards.

CHAPTER-7:

CONCLUSIONS AND RECOMMENDATION

7.1 **Outline**

This chapter concludes the report on the BankWeb project undertaken during the internship at Verizon Visible. It summarizes the key findings, limitations, and recommendations for future enhancements.

7.2 **Limitation/Constraints of the System**

Despite its success working, the BankWeb project has certain limitations and constraints that should be acknowledged:

- Limited scalability: The current architecture may face challenges in scaling to accommodate a large number of concurrent users or increased transaction volumes.
- Security vulnerabilities: While robust security measures are implemented, there may still be potential vulnerabilities that require further assessment and reinforcement.

7.3 **Future Enhancements**

To address the identified limitations and enhance the capabilities of the BankWeb application, the following recommendations for future enhancements are proposed:

- Scalability improvements: Implementing a more scalable architecture, such as microservices, could enhance the application's ability to handle increased loads and user traffic.
- Enhanced security measures: Continuous monitoring and updates to security protocols, including regular security audits and penetration testing, can help mitigate potential vulnerabilities and ensure data integrity.
- Integration of additional features: Incorporating additional functionalities such as bill payments, account alerts, and budgeting tools can enhance the value proposition of the application and improve user engagement.
- Performance optimization: Fine-tuning backend services and optimizing frontend components can improve application performance, resulting in faster response times and a smoother user experience.

7.4 **Inference**

The internship experience at Verizon Visible has been invaluable in providing hands-on exposure to real-world software development projects. Through the BankWeb project, I have gained practical experience in utilizing a diverse range of technologies, including React, Spring Boot, and JWT tokens, to develop a fully functional web application. Moreover, working collaboratively with the team at Verizon has enhanced my communication, problem-solving, and teamwork skills, preparing me for future roles in the software development industry. Overall, the internship has been a rewarding learning experience, and I look forward to applying the knowledge and skills acquired to contribute effectively to future projects.

APPENDIX

Microservices: Microservices is an architectural style that structures an application as a collection of loosely coupled, independently deployable services. Each service is responsible for a specific business capability and communicates with other services through APIs.

Full Stack Development: Full Stack Development refers to the development of both the front-end (client-side) and back-end (server-side) components of a web application. Full stack developers are proficient in multiple technologies and can work on all aspects of application development.

Single Sign-On (SSO): Single Sign-On is an authentication process that allows users to access multiple applications with a single set of login credentials. Once authenticated, users can navigate between different applications without the need to re-enter their credentials.

Authentication: Authentication is the process of verifying the identity of a user or system. It ensures that users are who they claim to be before granting access to resources or services.

Authorization: Authorization is the process of determining what actions users are allowed to perform within a system or application. It involves granting or denying access to specific resources based on the user's permissions.

RESTful API: RESTful API (Representational State Transfer) is an architectural style for designing networked applications. It uses standard HTTP methods (GET, POST, PUT, DELETE) to perform CRUD operations on resources, and follows principles such as statelessness and uniform interface.

JSON (JavaScript Object Notation): JSON is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate. It is commonly used for transmitting data between a server and a web application.

Frontend: Frontend refers to the client-side part of a web application, which users interact with directly. It includes components such as the user interface, design, and functionality that are visible to users.

Backend: Backend refers to the server-side part of a web application, which handles data storage, processing, and business logic. It interacts with the frontend and external services to fulfill client requests.

Deployment: Deployment is the process of making a software application available for use by installing it on servers or hosting platforms. It involves configuring the application, managing dependencies, and ensuring that it runs smoothly in a production environment.

H2 Database: H2 Database is an open-source, in-memory SQL database written in Java. It is lightweight, fast, and supports SQL syntax, making it suitable for development and testing purposes.

JWT (JSON Web Token): JWT is a compact, URL-safe means of representing claims to be transferred between two parties. It is commonly used for securely transmitting information between parties as a JSON object.

API (Application Programming Interface): API is a set of rules and protocols that allows different software applications to communicate with each other. It defines the methods and data formats that applications can use to request and exchange information.

REFERENCES

1. Verizon. Retrieved from https://en.wikipedia.org/wiki/Verizon
2. "Verizon Communications Inc. 2023 Annual Report (Form 10-K)". U.S. Securities and Exchange Commission. February 9, 2024. Retrieved from https://www.sec.gov/ix?doc=/Archives/edgar/data/732712/000073271224000010/vz-20231231.htm
3. React Documentation. Retrieved from https://react.dev/