DOCSCRIBE

A PROJECT REPORT

Submitted by

KESHVI PATEL (20BECE30161)

TEJAS PATEL (20BECE30188)

TIRTH PATEL (20BECE30190)

VISHAKHA RAJPUT (20BECE30219)

In fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

Computer Engineering

LDRP Institute of Technology and Research, Gandhinagar

Kadi Sarva Vishwavidyalaya

2023-2024

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE-IT Department

CERTIFICATE

This is to certify that the Project Work entitled **“DocScribe”** has been carried out by **Keshvi Patel (20BECE30161)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel	Dr. Sandip Modha
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE-IT Department

CERTIFICATE

This is to certify that the Project Work entitled **"DocScribe"** has been carried out by **Tejas Patel (20BECE30188)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel	Dr. Sandip Modha
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE-IT Department

CERTIFICATE

This is to certify that the Project Work entitled **“DocScribe”** has been carried out by **Tirth Patel (20BECE30190)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel	Dr. Sandip Modha
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE-IT Department

CERTIFICATE

This is to certify that the Project Work entitled **“DocScribe”** has been carried out by **Vishakha Rajput (20BECE30219)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2023-2024.

Prof. Hitesh Patel

Internal Guide

LDRP ITR

Dr. Sandip Modha

Head of the Department

LDRP ITR

Presentation-I for Project-III

1. Name & Signature of Internal Guide	
2. Comments from Panel Members	
3. Name & Signature of Panel Members	

ACKNOWLEDGEMENT

With immense pleasure we would like to present this report on our topic **DocScribe**. We are thankful to all that have helped us a lot for successful completion of our project and providing us courage for completing the work.

We are thankful to my Head of the Department **Dr. Sandip Modha**, our internal faculty guide **Prof. Hitesh Patel**, for providing guidance throughout our work giving us their valuable time.

Finally, We would like to thank our parents and friends. who have directly or indirectly helped us in making the project work successfully

With Regards,

Keshvi Patel (20BECE30161)
Tejas Patel (20BECE30188)
Tirth Patel (20BECE30190)
Vishakha Rajput (20BECE30219)

ABSTRACT

Docscribe is a state-of-the-art web application crafted with Next.js, React, Drizzle, Shadcn UI, Superbase, Tailwind, and Socket.io, ushering in a new era of collaborative document management. This project seamlessly transforms design concepts into a fully functional landing page while offering users the freedom to choose between dark and light modes for a personalized experience.

The application boasts a robust authentication system, encompassing login, signup, two-factor authentication, and custom email confirmation templates, all powered by Superbase. Upon signup, users are prompted to create workspaces, setting the stage for a personalized document organization journey.

Docscribe introduces a tiered subscription model, distinguishing between free and pro plans. Pro plan members enjoy superior features, including custom logos, extended collaborators, and increased workspace capacity. Real-time plan utilization feedback prompts free plan members to seamlessly upgrade when exceeding limitations.Collaboration lies at the heart of Docscribe, shaping its organizational structure around workspaces, folders, and files. Workspaces can be configured as private, shared, or collaborating, offering flexibility in document collaboration. Real-time updates permeate the application, facilitating instant changes across the entire interface.

A unique feature of Docscribe is the implementation of real-time cursors, providing a visual representation of collaborators' actions within documents. Superbase presence enhances collaboration awareness by showcasing collaborating users and their activities.

Docscribe not only embraces cutting-edge technologies but also introduces innovative features like real-time cursors and presence, positioning itself as a forward-thinking solution for collaborative document creation and management.

TABLE OF CONTENT

Acknowledgement **i**

Abstract **ii**

Table of content **iii**

Table of figures **v**

1 Introduction **1**

1.1 Introduction 2

1.2 Aims and Objective of the work 2

1.3 Brief Literature Review 2

1.4 Problem definition 3

1.5 Plan of their work 3

2 Technology and Literature Review 4

2.1 About Tools and Technology 5

2.2 Brief History of Work Done 5

3 System Requirements Study **6**

3.1 User characteristics 7

3.2 Hardware & Software Requirements 7

3.3 Assumption and dependencies 8

4 System Diagrams **10**

4.1 Design Goal 11

4.2 Class Diagram 12

4.3 Data flow Diagram 13

4.4 State diagram / UML diagram 15

4.5 Activity Diagram 16

4.6 Sequence Diagram 17

5 Data Dictionary 18

6 Testing 20

6.1 Unit testing 22

6.2 Integration testing 22

6.3 Functional testing 22

6.4 Security testing 22

6.5 User interface testing 23

7 Result, Discussion and Conclusion 24

7.1 Result 25

7.2 Discussion 25

7.3 Conclusion 25

8 Output Screenshot 26

9 Biblography 30

TABLE OF FIGURES

Figure no.	Figure Name	Page no.
4.1	Class diagram	12
4.3	DFD level-0 diagram	13
4.4	DFD level-1 diagram	14
4.5	UML Diagram	15
4.6	Activity diagram	16
4.7	Sequence Diagram	17

Chapter1-Introduction

1.1. **Introduction**

1.2. **Aims and Objective of the work**

1.3. **Brief Literature Review**

1.4. **Problem definition**

1.5. **Plan of their work**

1.1. Introduction

- DocScribe is an all-in-one workspace and collaborative platform that allows individuals and teams to organize their work, information, and projects in a centralized and flexible manner.
- It has an ability to replace multiple productivity tools with a single, cohesive platform, providing a unified workspace for a wide range of tasks.
- Whether it's co-authoring documents, managing projects, or providing feedback, DocScribe's collaborative tools enhance communication and productivity in real-time.

1.2. Aims and Objective of the work

- Docscribe aims to revolutionize collaborative document management allowing users to create, edit, and manage documents collaboratively in real-time, emphasis on security, implementing a robust authentication system that includes login, signup, and two-factor authentication to safeguard user data.
- The primary objectives include enhancing real-time collaboration with features like cursors and Superbase presence, ensuring a user-friendly design with options for dark and light modes, implementing a robust authentication system, introducing a tiered subscription model for diverse user needs, ensuring a responsive design for a consistent experience, facilitating real-time updates and user feedback, optimizing document organization with versatile workspaces, integrating Stripe for seamless subscription management, introducing innovative features like real-time cursors, and empowering users with customization options for a personalized experience. Docscribe seeks to redefine the landscape of document collaboration through innovation, efficiency, and user empowerment.

1.3. Brief Literature Review

- Contemporary literature highlights the evolution of collaborative document management, emphasizing the necessity of real-time collaboration features and user-centric design principles. Studies underscore the importance of robust authentication systems, incorporating multi-factor authentication for heightened security.
- The adoption of tiered subscription models aligns with emerging trends in accommodating diverse user needs. Integration with third-party services, like Stripe, reflects literature-backed strategies for seamless financial transactions within applications.
- Additionally, the literature recognizes the value of innovative features such as real-time cursors in enhancing collaboration awareness. Responsive design across devices remains a key consideration, acknowledging the increasing reliance on mobile platforms for document collaboration. These insights guide the development of Docscribe, ensuring alignment with contemporary best practices

1.4. Problem definition

- Traditional document management tools often lack the agility and real-time collaboration features required in today's dynamic work environments. Users face challenges in seamless collaboration, personalized document organization, and secure authentication.
- Existing platforms may not fully address the varying needs of users through tiered subscription models. Additionally, the absence of innovative features like real-time cursors hinders efficient collaboration. Docscribe aims to address these issues by introducing a cutting-edge document management solution that prioritizes real-time collaboration, user-centric design, robust security measures, and a versatile subscription model to meet the diverse requirements of users.

1.5. Plan of their work

- The docScribe project encompasses a journey of designing and developing a full-stack Notion clone with real-time collaboration, rich text editing capabilities, and subscription management. In the first month, we'll meticulously plan and design the application, defining user stories, wireframing UI layouts, and structuring the database schema. Subsequently, we'll delve into backend development over the next two months, leveraging technologies like Next.js and Supabase to establish robust backend infrastructure and RESTful APIs.
- Frontend development will then take precedence as we craft intuitive user interfaces and integrate real-time collaboration features using Websockets, alongside a custom rich text editor and responsive design elements. Months four and five will be dedicated to implementing subscription management functionalities, integrating payment gateways, and ensuring seamless user experiences in managing subscriptions and billing.
- Finally, the last month will focus on comprehensive testing, deployment, and documentation to deliver a polished, scalable, and user-centric application ready for production.

Chapter2-Technology and Literature Review

2.1. About Tools and Technology

2.2. Brief History of Work Done

2.1. About Tools and Technology:

- **Operating System:** Any version of Windows family(4.0&above)
- **Frontend:** NextJs, React, Tailwind, Shadcn
- **Orm:** Drizzle
- **Backend:** Supabase, Stripe , Socket io, NextJs

2.2. Brief History of Work Done

- **Planning of the application:** This includes determining the features that the application will need, as well as the user interface and data model. It will be answering the questions like what type of features will the application needs.What can be the user interface look like?What data will the application need to store?How will the application work?
- **Designing a user interface:** One planning of the application is done,we need to design the user interface.This involves creating the NextJs, Tailwind CSS and React that will be used to display the application's interface.The user interface Should be easy to use and navigate. It should also be visually appealing.
- **Developing the backend logic:**The backend logic is the code that will be used to handle the application's business logic.This includes code for generating invoices, sending invoices to backend, and storing the details of invoices. Along with it, seeming theuser's profile information and providing authentication support to the Application.
- **Testing the application:**Once the application is developed, we need to test it to ensure that it works correctly.TI1 is involves testing the application with different Scenarios.We need to also test the application for security vulnerabilities.
- **Deploying the application:**Once the application istested,we need to deploy it so that users canaccess it. Itinvolves making the application available on a web server.We have to make sure that the application is secure and that it can handle a high volume of traffic.

Chapter 3 – System Requirements Study

3.1. User Characteristics

3.2. Hardware and Software Requirements

3.3. Assumptions and Dependencies

3.1. User Characteristics

Analyzing user characteristics is an important aspect of any project. It allows us to clearly define and focus on who the end users are for the project. Also, it allows checking the progress of the project to ensure thatwe are still developing the system forthe endusers. Tue user must have following characte1istics:

- User must have basic knowledge of computers.
- User should understand the use of all modules.
- User can easily interact with the proposed system.
- User must know the method to upload an image in the app.
- User should be also being aware about the running process of the system.

3.2. Hardware and Software Requirements

Software and hardware requirements are used to desc1ibethe minimum hardware and software requirements to nm the software. These requirements are desc1ibed below:

Hardware Requirements

- **Client:**
    - 256MBofRAM
    - 1.6 GHz CPU
    - InternetConnection
    - Monitor
    - Keyboard/Mouse
    - Printer
    - GPS
- **Server:**
    - 1.6 GHzCPU
    - 1GBofRAM
    - InternetConnection
    - Monitor
    - Keyboard/Mouse
    - Printer

Software Requirements

- Client:
  - OperatingSystem:WindowsorLinux,AndroidoriOS
  - Web Browser: any HTMLcompliantbrowser,GoogleChrome (HighlyPreferred)
  - Node.js version 18.3.1
  - Languages: Multiple Languages
  - License: Open Source
  - Packages:React,Next,Node,TailwindCSS
- Server:
  - OperatingSystem:Windows orLinux,Androidor iOS
  - Database: MongoDB
  - Technologies:HTML,TailwindCSS,Typescript

3.3. Assumptions and Dependencies

3.4.1 Assumptions

- The Docscribe project operates under several key assumptions to provide an optimal user experience. It assumes users have consistent internet connectivity for real-time collaboration and access to the application.

- Browser compatibility is assumed, expecting users to utilize modern browsers compatible with technologies like Next.js, React, and Socket.io. The implemented security measures, including the authentication system and data encryption, are assumed to effectively safeguard user data and maintain collaboration confidentiality.

- Users are expected to possess a basic understanding of document management concepts and the collaborative tools provided by the application. The project also assumes the continuous availability and proper functioning of third-party services, particularly Superbase and Stripe, for database management and subscription handling.

3.4.2. Dependencies

- DocScribe is dependent on the availability of the internet for all users to access the application successfully. Users must have a stable and reliable internet connection to interact with the platform, browse property listings, and make bookings.

- Users must enter valid login credentials (username and password) to access DocScribe and perform further actions. If users provide incorrect or invalid login information, the system will not grant access.

- The efficiency of DocScribe's performance and data handling is crucial for its successful operation. The application's dependency on efficient data retrieval, database management, and code optimization ensures a smooth and responsive user experience.

Chapter 4 – System Analysis

4.1 Design Goal

4.2 Class Diagram

4.3DFD level-0 Diagram

4.4 DFD level-1 Diagram

4.5 UML diagram

4.6 Activity Diagram

4.7 Sequence Diagram

4.1 Design Goal

- The design goal of Docscribe is to create an innovative and user-centric collaborative document management platform that seamlessly integrates advanced technologies. The primary objectives include transforming a meticulously designed interface into a fully functional landing page, providing users with the choice between dark and light modes for a personalized experience

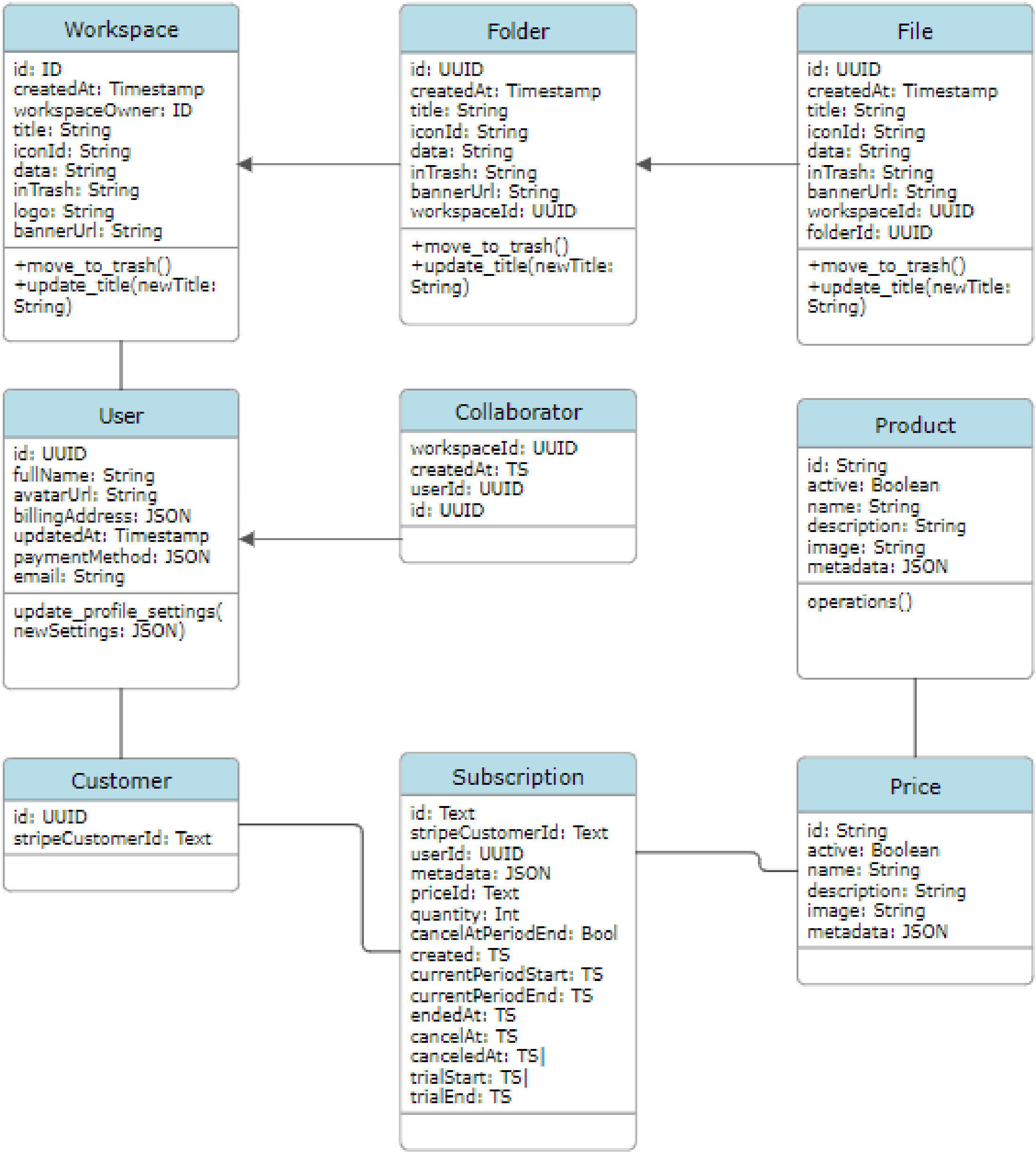
- The platform aims to establish a robust authentication system, leveraging Superbase for secure login, sign-up, and two-factor email confirmation with a custom email template. Docscribe strives to enhance user collaboration by introducing workspaces, folders, and files, each offering private, shared, or collaborative settings.

- The design prioritizes real-time updates and responsiveness, enabling users to make changes across the application client-side in real time. It also aims to introduce unique features such as the ability to customize icons and banners for folders and files, showcasing immediate feedback and real-time collaboration.

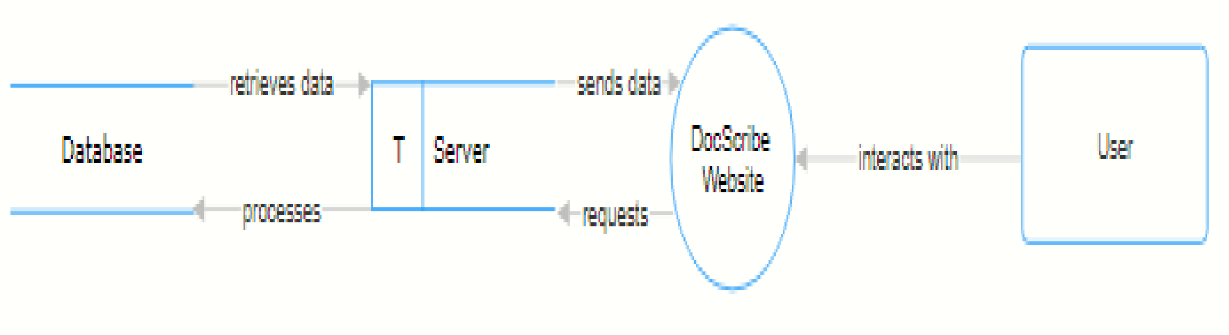
- The implementation of a tiered subscription model, facilitated by Stripe, caters to both free and Pro Plan members, offering distinct features based on their subscription level. Real-time cursors and presence, as well as the capability to restore deleted files and manage workspace settings, contribute to a comprehensive and collaborative user experience.

- The design goal encompasses the creation of an intuitive, secure, and feature-rich platform that revolutionizes how users collaborate on documents.

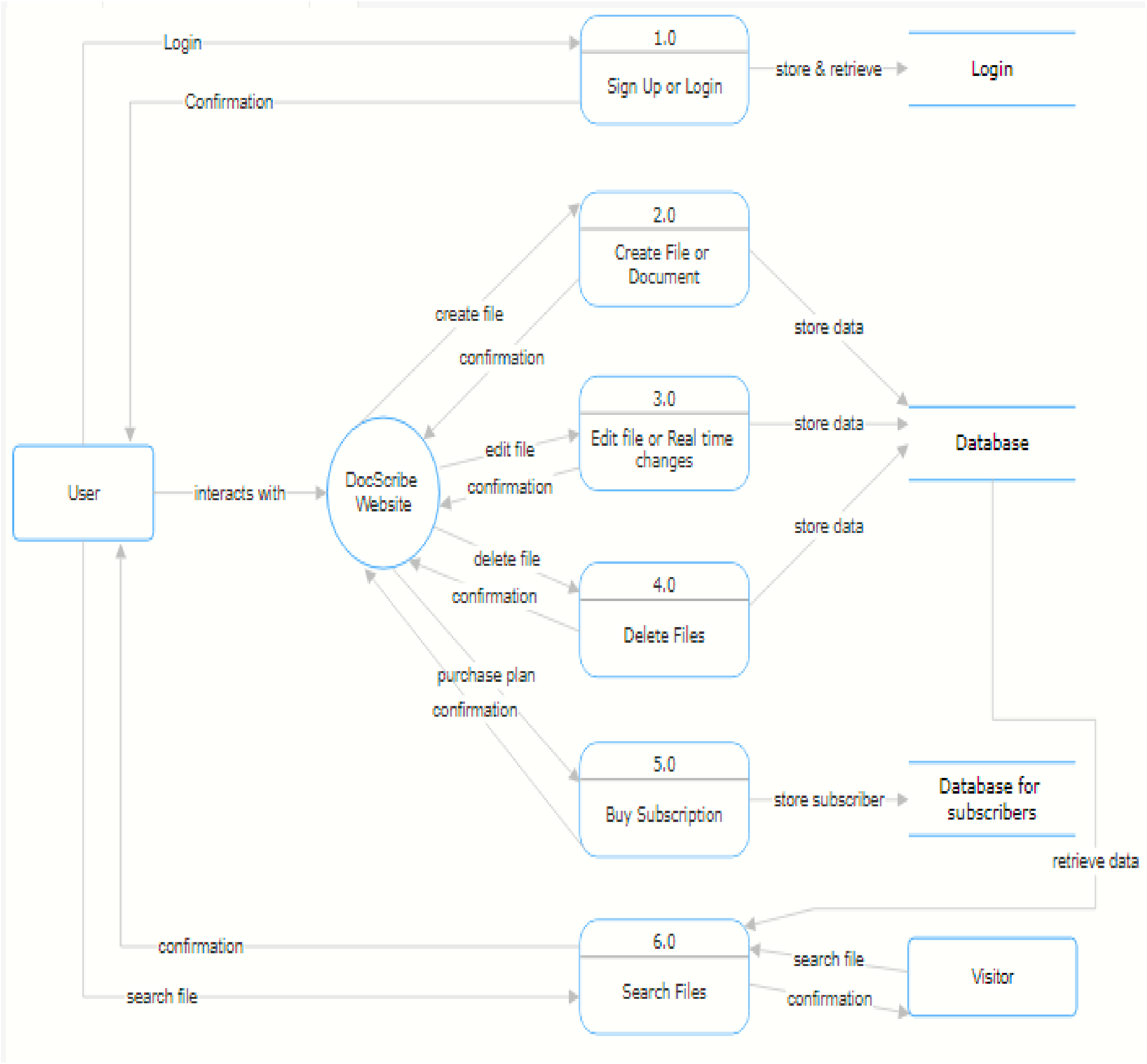
4.2 Class Diagram



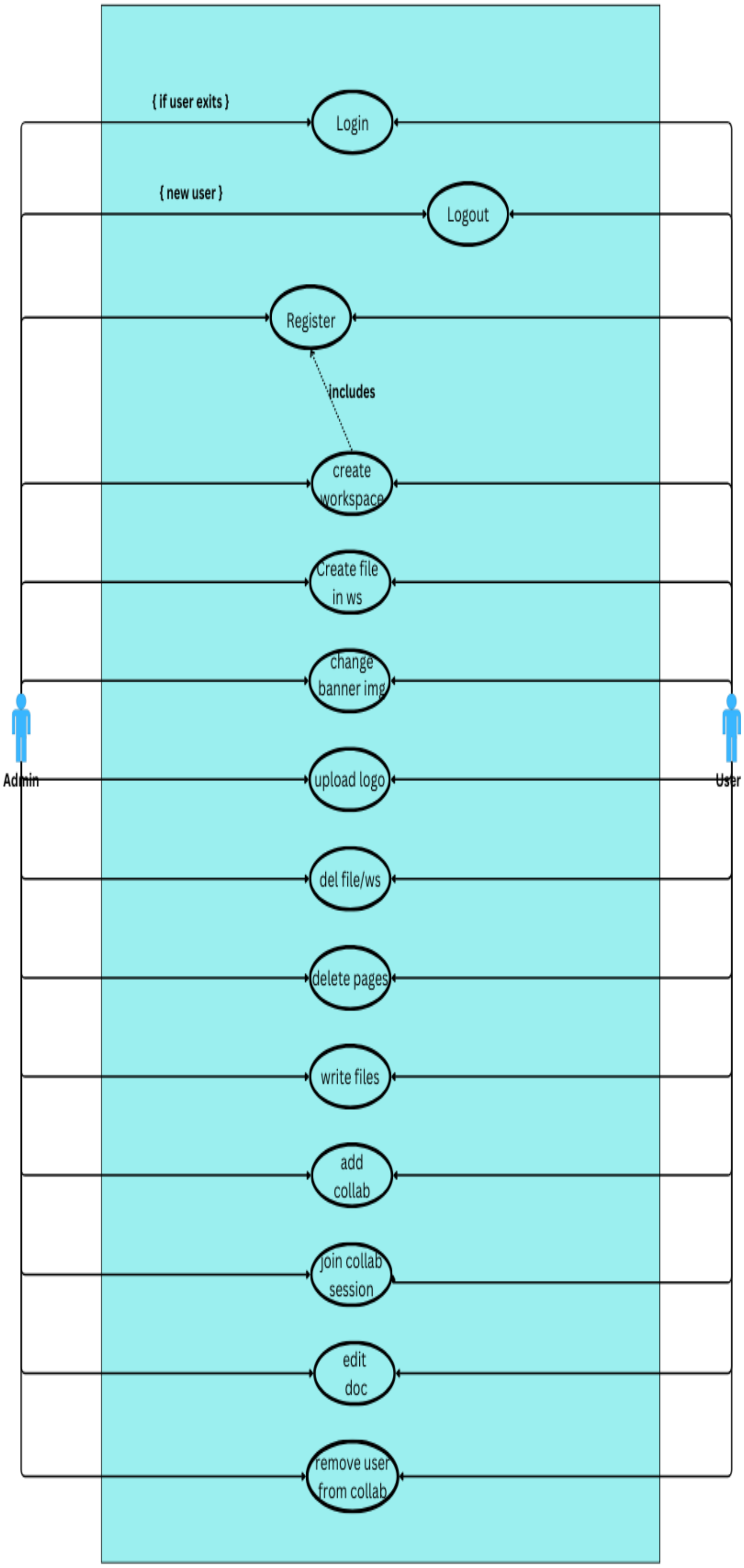
4.3 DFD level-0 Diagram



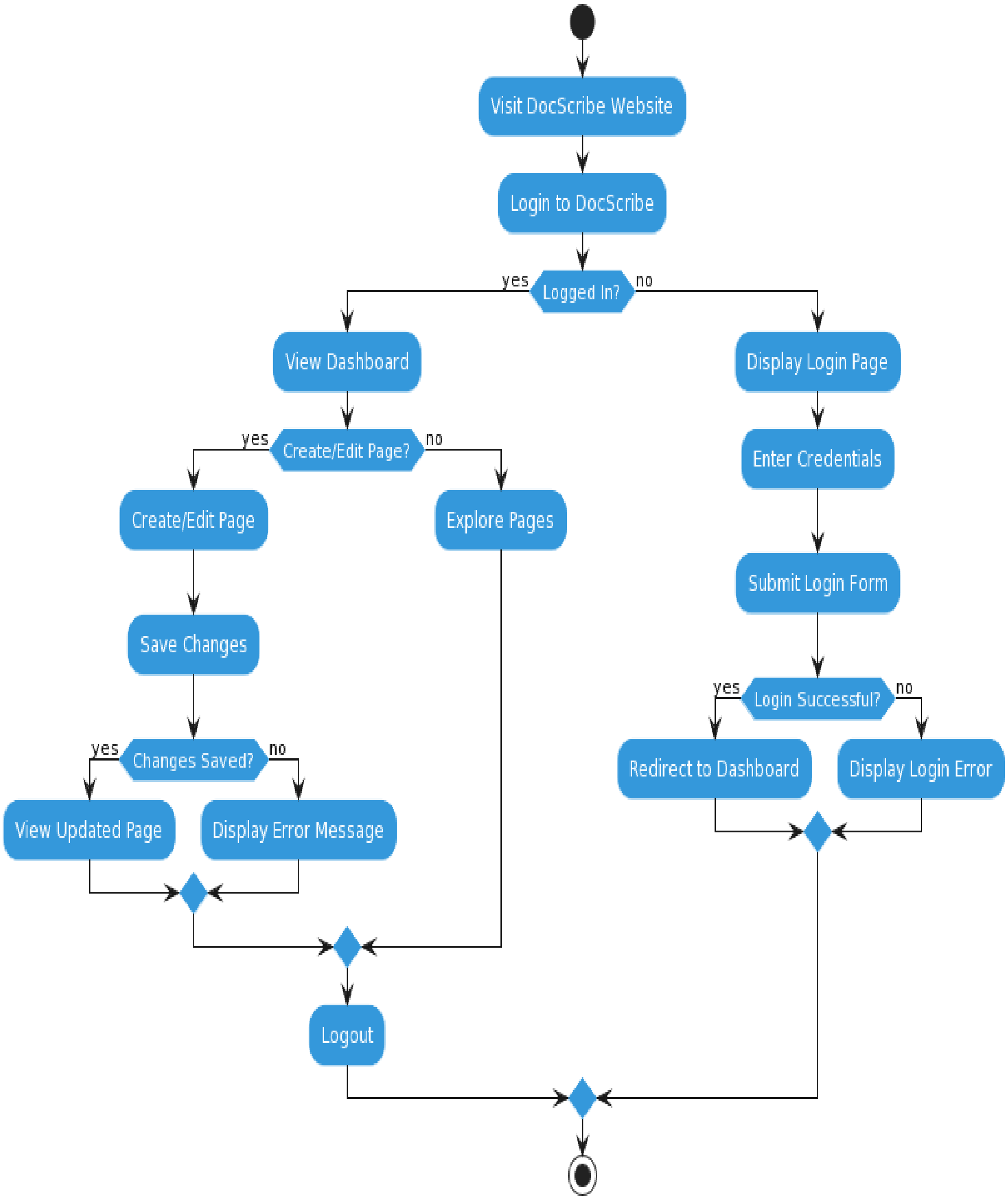
4.4 DFD level-1 Diagram



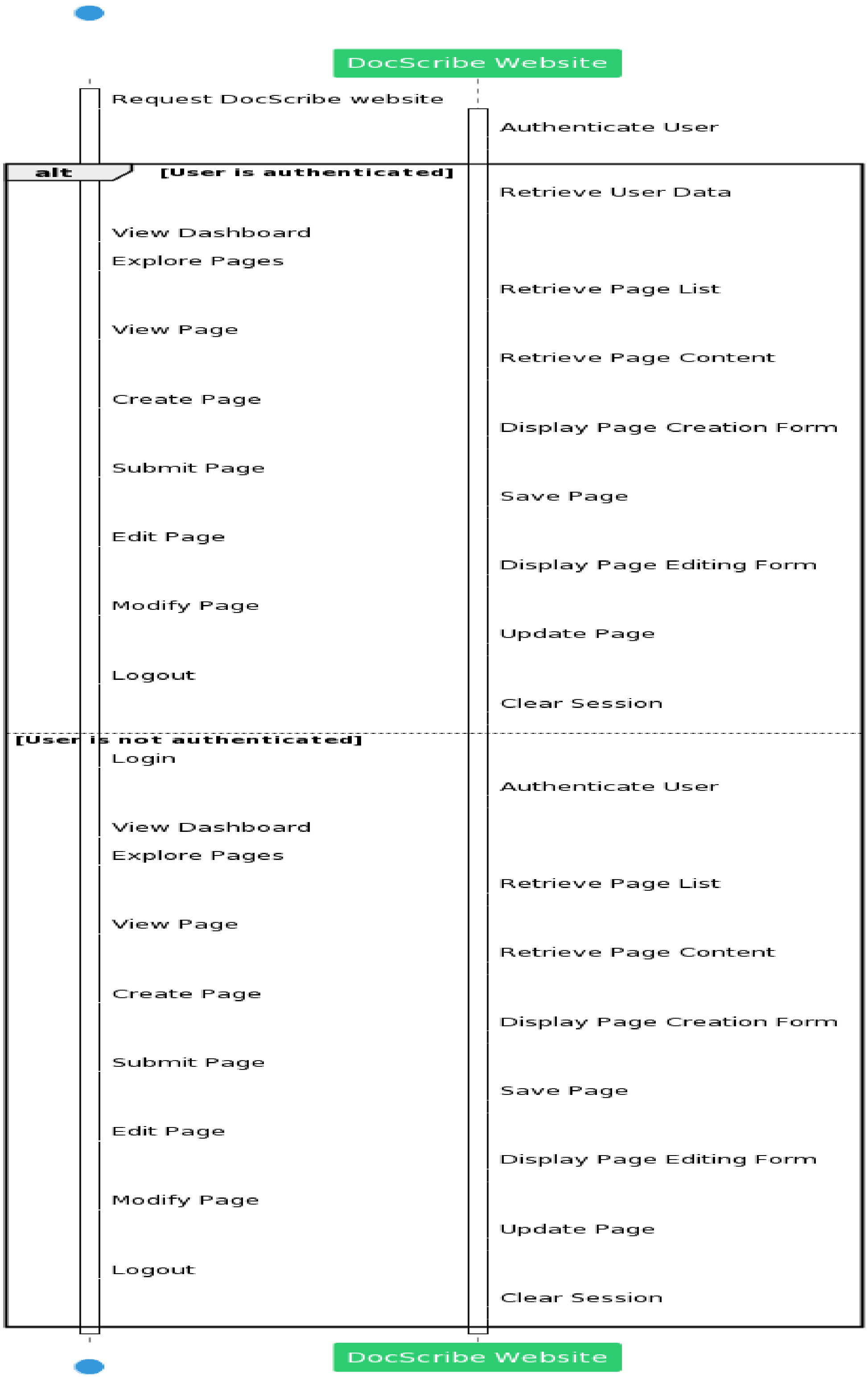
4.5 State Transition / UML Diagram



4.6 Activity Diagram

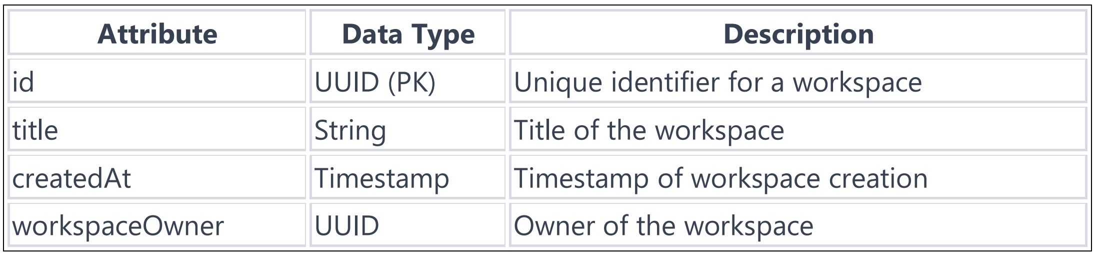


4.7 Sequence Diagram



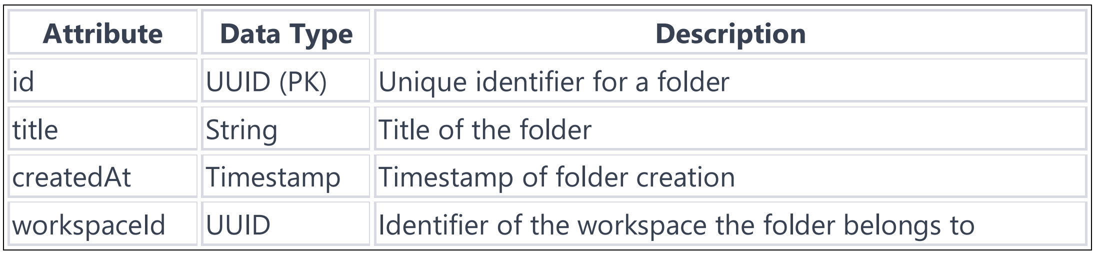
Chapter 5 – Data Dictionary

WorkSpace



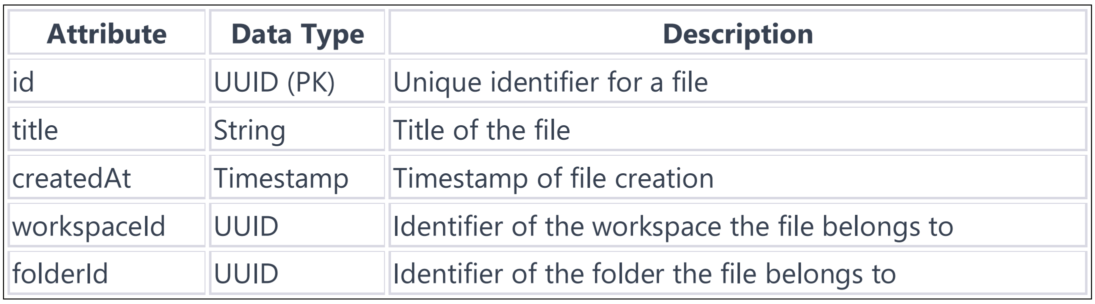
Attribute	Data Type	Description
id	UUID (PK)	Unique identifier for a workspace
title	String	Title of the workspace
createdAt	Timestamp	Timestamp of workspace creation
workspaceOwner	UUID	Owner of the workspace

Folder



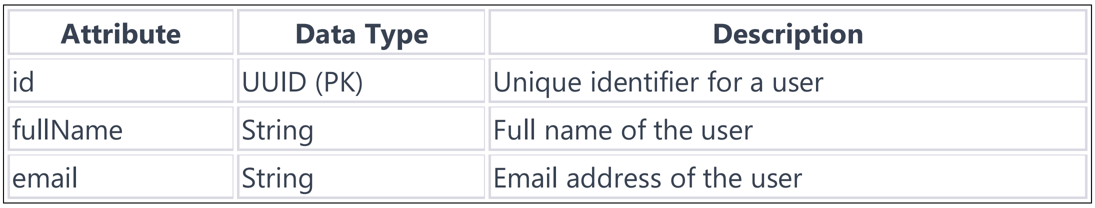
Attribute	Data Type	Description
id	UUID (PK)	Unique identifier for a folder
title	String	Title of the folder
createdAt	Timestamp	Timestamp of folder creation
workspaceId	UUID	Identifier of the workspace the folder belongs to

File



Attribute	Data Type	Description
id	UUID (PK)	Unique identifier for a file
title	String	Title of the file
createdAt	Timestamp	Timestamp of file creation
workspaceId	UUID	Identifier of the workspace the file belongs to
folderId	UUID	Identifier of the folder the file belongs to

User



Attribute	Data Type	Description
id	UUID (PK)	Unique identifier for a user
fullName	String	Full name of the user
email	String	Email address of the user

Collaborator

Attribute	Data Type	Description
workspaceId	UUID	Identifier of the associated workspace
userId	UUID	Identifier of the associated user
id	UUID (PK)	Unique identifier for a collaborator

Customer

Attribute	Data Type	Description
id	UUID (PK)	Unique identifier for a customer
stripeCustomerId	String	Identifier for the customer in Stripe

Subscription

Attribute	Data Type	Description
id	String (PK)	Unique identifier for a subscription
userId	UUID	Identifier of the associated user
priceId	String	Identifier of the associated price

Price

Attribute	Data Type	Description
id	String (PK)	Unique identifier for a price
productId	String	Identifier of the associated product
unitAmount	Number	Amount of the price unit
currency	String	Currency of the price

Product

Attribute	Data Type	Description
id	String (PK)	Unique identifier for a product
name	String	Name of the product

Chapter 6 – Testing

6.1 Unit Testing

6.2 Integration Testing

6.3 Functional Testing

6.4 Security Testing

6.5 UI Testing

➢ Testing

- The motivation behind testing is to find mistakes. Testing is the way toward attempting to find each possible deficiency or shortcoming in a work item. It gives an approach to check the usefulness of parts, sub congregations, gatherings and additionally a completed item it is the way toward practicing programming with the goal of guaranteeing that the Software framework lives up to its prerequisites and user desires and does not bomb in an unsuitable way. There are different kinds of test. Each test type tends to a particular testing necessity.

6.1 Unit Testing

- Unit testing involves the evaluation of individual components or modules to ensure that they function as intended. In the context of Docscribe, this would mean verifying that each specific function or module, such as authentication processes or real-time updates, performs accurately. Tools like Jest and React Testing Library can be utilized to execute these tests, allowing developers to catch and rectify issues at an early stage of development.

6.2 Integration Testing

- As Docscribe integrates various components, services, and modules, integration testing becomes crucial. This form of testing evaluates how different parts of the application interact with each other. Ensuring seamless collaboration between, for example, the authentication system and the real-time collaboration features is vital for a cohesive user experience. Jest and React Testing Library, alongside tools like Supertest for backend services, can be employed for effective integration testing.

6.3 Functional Testing

- Functional testing focuses on the end-to-end evaluation of application features and user interactions. This ensures that the application behaves as expected from the user's standpoint. For Docscribe, functional testing will validate that features like creating workspaces, collaborating in real-time, and managing subscriptions work seamlessly. Cypress and Selenium are well-suited for implementing these comprehensive tests.

6.4 Security Testing

- The security of Docscribe is paramount, and security testing aims to identify and rectify potential vulnerabilities. Tools like OWASP ZAP and SonarQube can be employed to conduct thorough security assessments, ensuring that the application is resilient against common security threats such as injection attacks, cross-site scripting (XSS), and other potential vulnerabilities.

6.5 User Interface Testing

- UI testing assesses the visual elements and user interface to guarantee a user-friendly and visually consistent experience. Docscribe, with its emphasis on responsive design and customizable interfaces, benefits greatly from UI testing. Using tools like Cypress and React Testing Library enables developers to validate that UI components render correctly and respond appropriately to user interactions.

Chapter 7 – Result, Discussion and Conclusion

7.1 Result

7.2 Discussion

7.3 Conclusion

7.1 Result

- The demonstration of the Docscribe project showcased a successful integration of cutting-edge technologies, including Next.js 13, React, Drizzle ORM, Shadcn UI (assumed to be Tailwind CSS), Socket.io, Superbase, and Stripe.
- The implementation of real-time collaboration features, such as cursors and Superbase presence, demonstrated synchronized editing and efficient communication between collaborators.
- The user interface, featuring dark and light modes, reflected a user-centric design approach, providing a seamless and customizable experience. The integration of Stripe for subscription payments and the handling of financial transactions demonstrated a robust and secure backend.
- Real-time updates, dynamic folder and file creation, and innovative features like real-time cursors added significant value to the overall user experience.

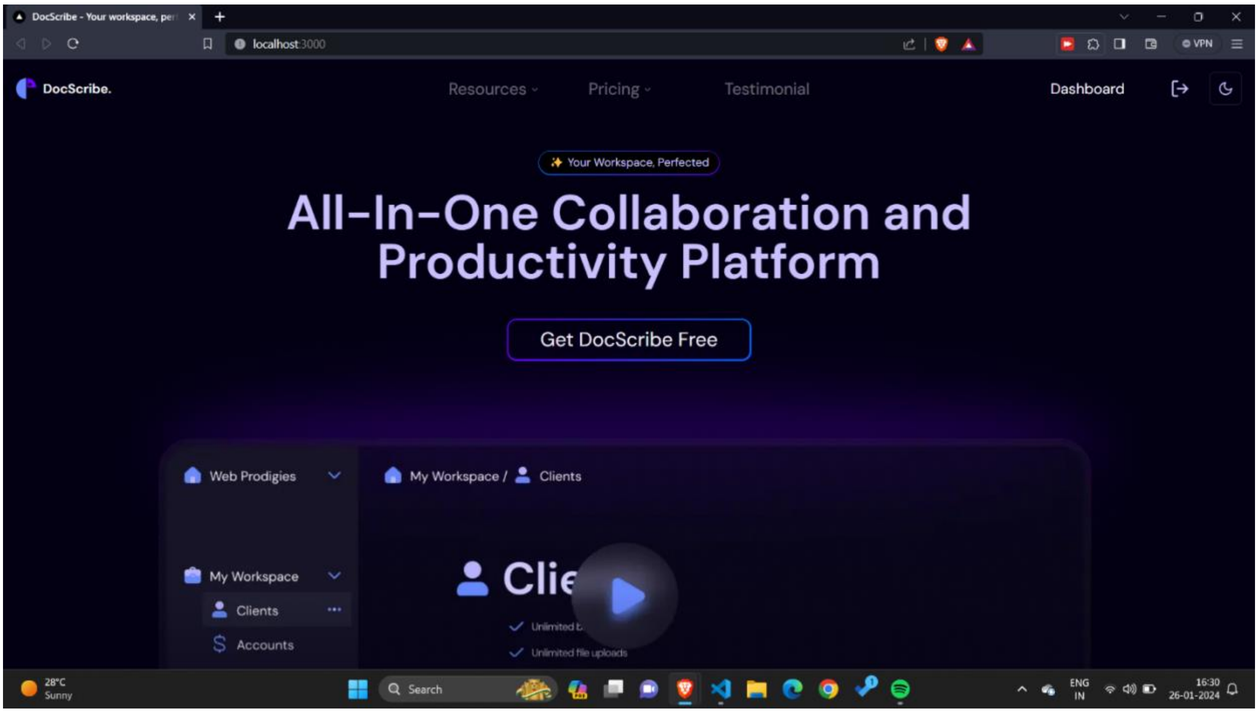
7.2 Discussion

- The project discussion revolves around the successful realization of key features. The tiered subscription model, distinguishing between free and Pro Plan members, offers flexibility and caters to varying user needs.
- The implementation of real-time cursors, a feature rarely seen on YouTube, introduces an unprecedented level of collaboration awareness, enhancing user interaction within documents. The application's responsiveness across mobile and desktop platforms ensures a consistent experience for users regardless of their device.
- Collaborative functionalities, such as restoring deleted files and seamless billing updates, were highlighted as significant strengths during the demonstration. The challenges embedded in the project invite further engagement and exploration, fostering continuous improvement.

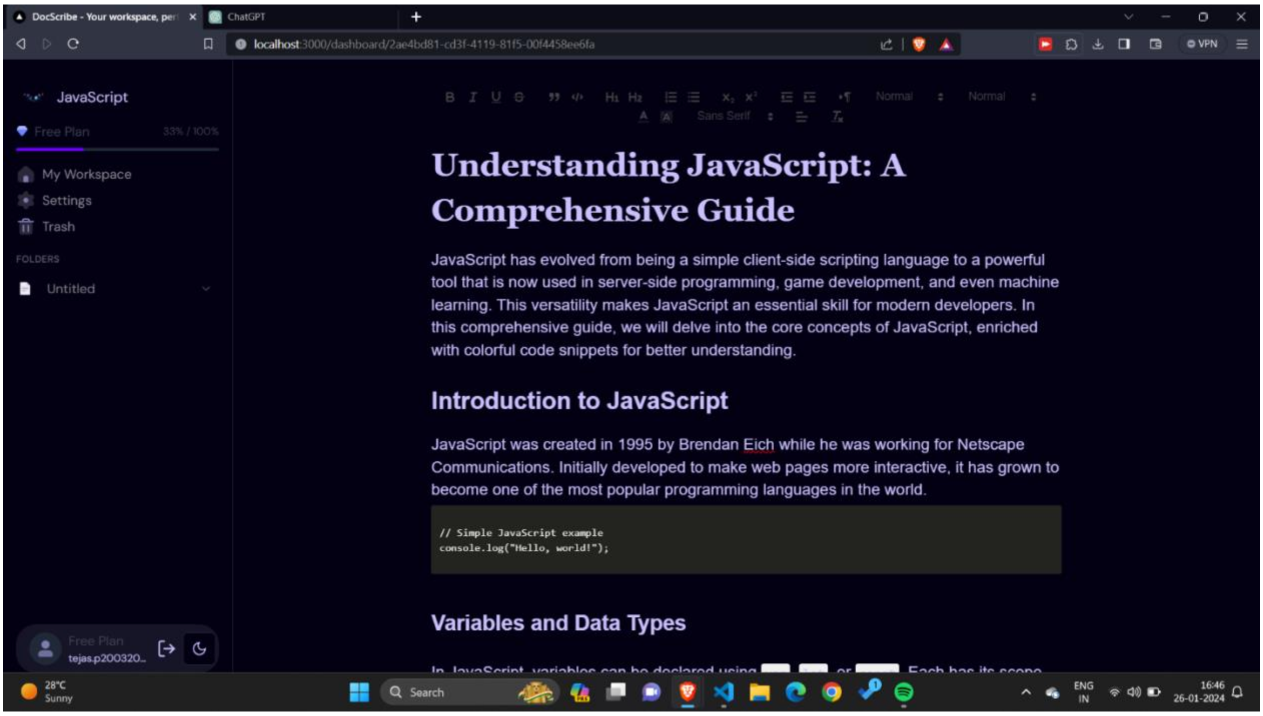
7.3 Conclusion

- In conclusion, Docscribe emerges as a highly promising and innovative solution in the realm of collaborative document management. The successful integration of technologies and the demonstration of advanced features underscore its potential to redefine user collaboration experiences.
- The project's responsiveness, security measures, and real-time capabilities contribute to its robustness and reliability. As it progresses towards deployment, Docscribe holds the promise of becoming a cornerstone in modern document collaboration platforms, offering a seamless, secure, and feature-rich environment for users to create, collaborate, and manage documents.
- The challenges set forth in the project provide avenues for continued development and enhancement, ensuring that Docscribe remains at the forefront of collaborative tools.

Chapter 8 – Output Screenshots



All-In-One Collaboration and Productivity Platform



Understanding JavaScript: A Comprehensive Guide

JavaScript has evolved from being a simple client-side scripting language to a powerful tool that is now used in server-side programming, game development, and even machine learning. This versatility makes JavaScript an essential skill for modern developers. In this comprehensive guide, we will delve into the core concepts of JavaScript, enriched with colorful code snippets for better understanding.

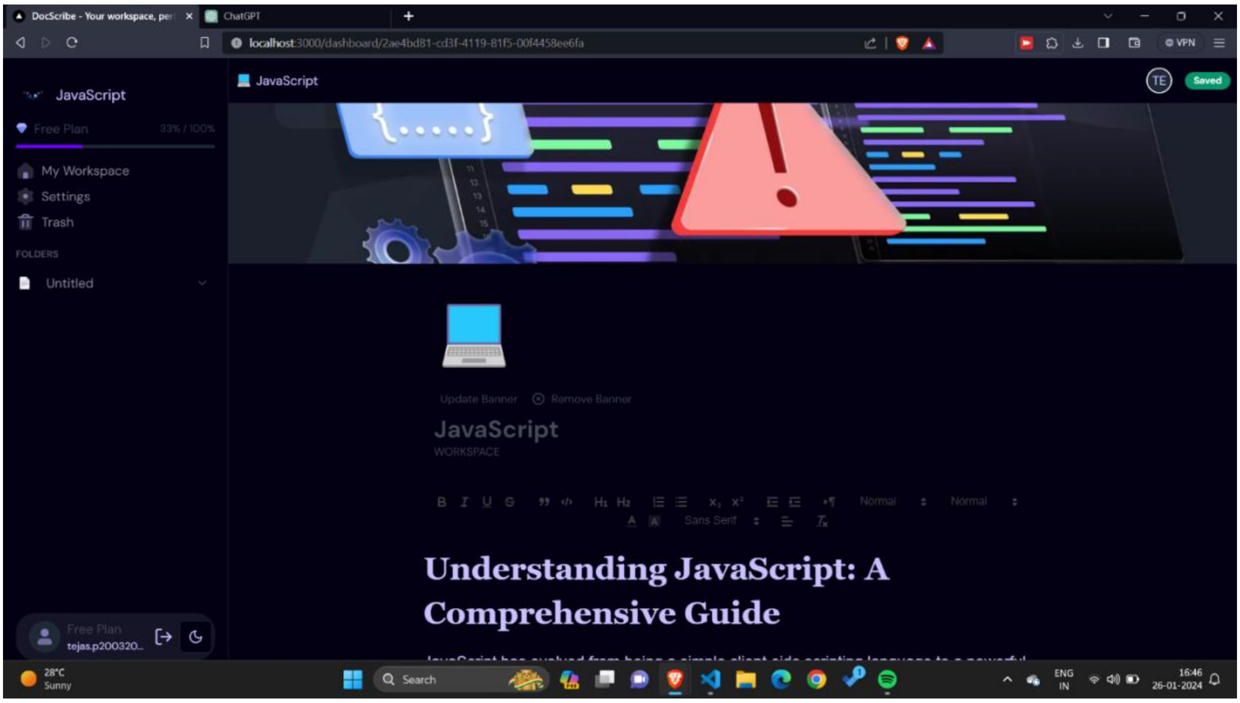
Introduction to JavaScript

JavaScript was created in 1995 by Brendan Eich while he was working for Netscape Communications. Initially developed to make web pages more interactive, it has grown to become one of the most popular programming languages in the world.

```
// Simple JavaScript example
console.log("Hello, world!");
```

Variables and Data Types

In JavaScript, variables can be declared using [illegible], [illegible] or [illegible]. Each has its scope



Understanding JavaScript: A Comprehensive Guide

Chapter 9 – Bibliography

Bibliography

- https://tailwindcss.com/docs/installation
- https://nextjs.org/docs
- https://ui.shadcn.com/docs
- https://socket.io/
- https://orm.drizzle.team/
- https://supabase.com/