

Experiment No 9

Trust Docs

Block Chain Mini Project Report

Submitted in partial fulfilment of the requirement of University of Mumbai For

The degree of (computer engineering)

By

Roll No.	Name
A56	Mayur Argade
A45	Sumedh Gamare
A63	Aniket Tagadkar
A	Hemant Khutmode

Under the Guidance of

Prof. Smita Kulkarni-Pai



Department of Computer Engineering

TERNA ENGINEERING COLLEGE

Plot no.12, Sector-22, Opp. Nerul Railway station,

Phase-11, Nerul (w), Navi Mumbai 400706

UNIVERSITY OF MUMBAI

Index

SR. No	Content	Page No
1	Abstract	3
2	Introduction	4
3	Flow Diagram	5
4	Sequence Diagram	6
5	Technology Used	7
6	Code	8
7	Output	12
8	Conclusion	14

Chapter 1: Abstract

Trust Docs: Revolutionizing Document Verification with Blockchain Technology

- **Immutable Verification:** Trust Docs leverages cutting-edge blockchain technology to establish an innovative method of document verification. The system's foundation lies in recording the hash values of user-uploaded documents on an unalterable ledger. This ensures that the documents remain impervious to tampering, safeguarding their authenticity.
- **Blockchain Security:** The heart of Trust Docs beats with the security provided by blockchain. Once a document's hash is entered into the ledger, it becomes etched in the blockchain's unchangeable history. This intrinsic feature significantly mitigates the risk of fraud and other illicit activities.

- **User Authentication:** Authenticated users enjoy the privilege of not only verifying documents on their local machines but also maintaining a comprehensive record of their uploaded documents. This adds a layer of transparency and accountability to the document verification process.
- **Accessibility for All:** Trust Docs is designed to cater to a wide audience. Unauthenticated users can efficiently verify the authenticity of documents, enhancing the platform
- **Tamper-Proof Records:** The use of blockchain technology ensures that once a document is recorded on the ledger, its hash values cannot be modified. This tamper-proof characteristic adds an extra layer of reliability, instilling confidence.

Chapter 2: Introduction

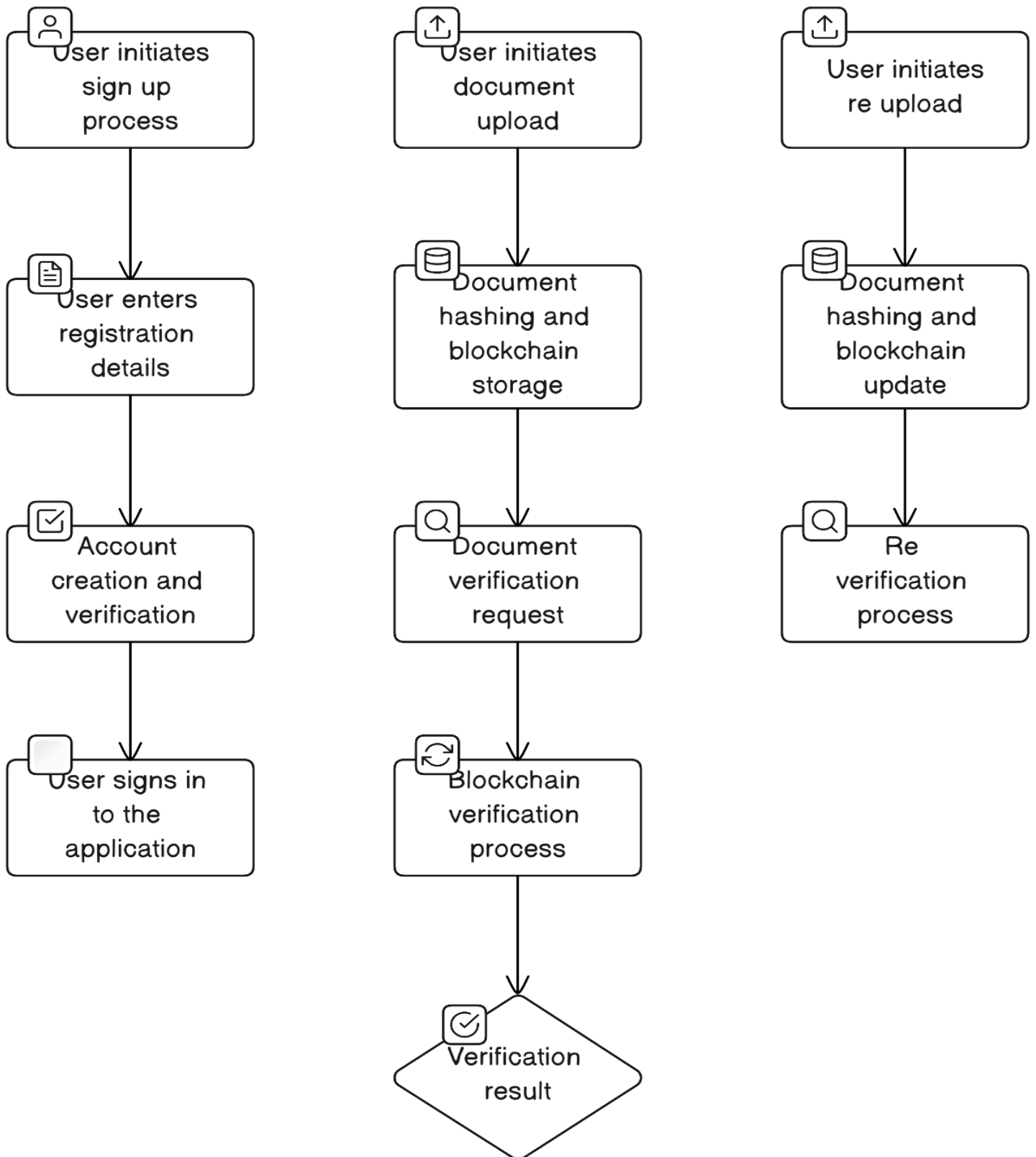
In the ever-evolving digital landscape, the need to safeguard the integrity and authenticity of documents has become increasingly crucial. Traditional document verification methods often fall prey to vulnerabilities such as forgery and manipulation. To address these concerns and provide a secure, transparent, and tamper-proof solution, we proudly introduce **TrustDocs**: a revolutionary web application that harnesses the power of blockchain technology to establish a robust document verification system.

TrustDocs empowers users to upload their documents and have their document hashes stored on the blockchain ledger, creating an immutable record that cannot be altered or forged. This innovative solution leverages popular technologies such as React for the front-end, Node/Express for the back-

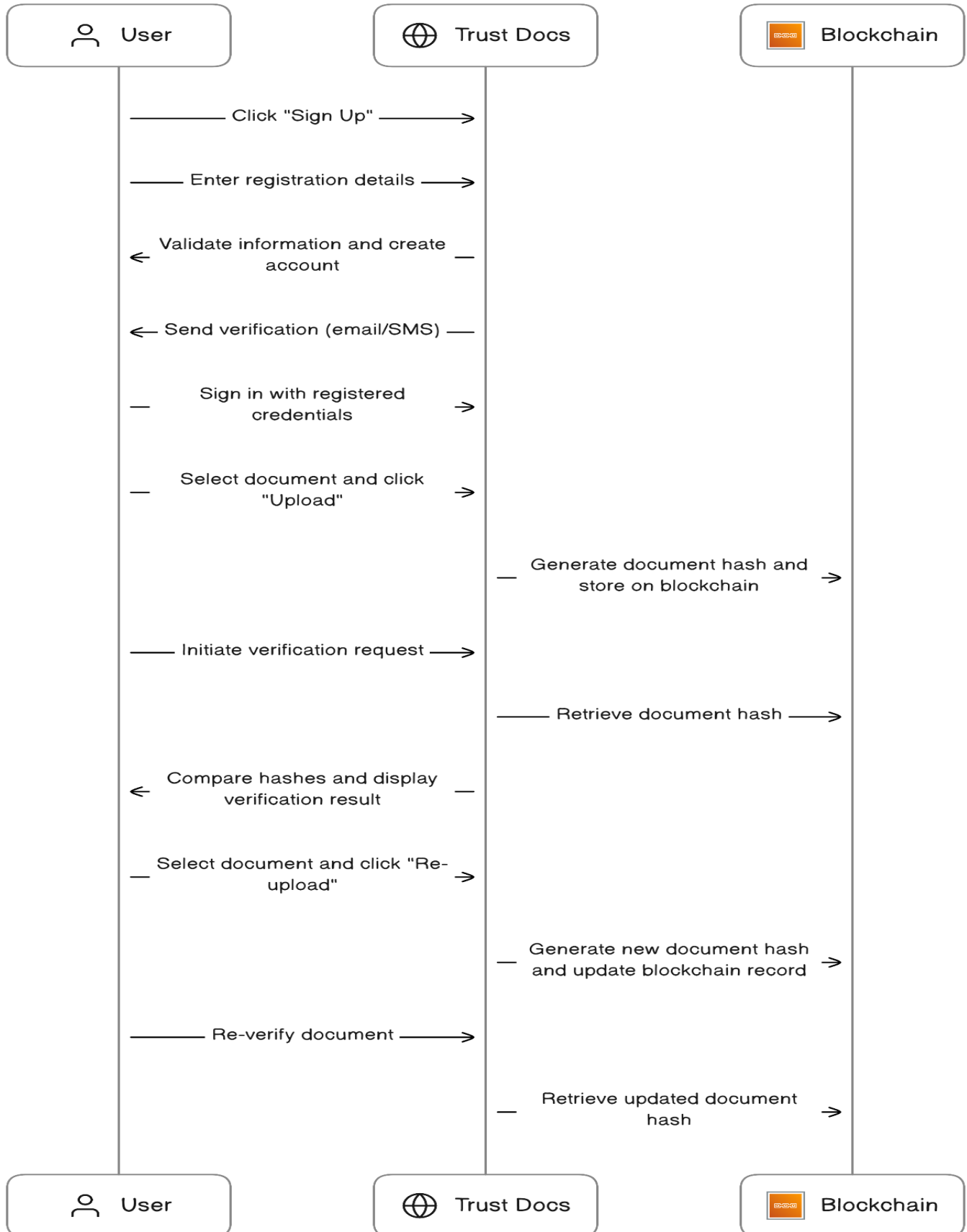
end, and MongoDB for data storage. During the initial development phases, we utilize Ganache, a local blockchain, to ensure a seamless and efficient development process.

With **TrustDocs**, users can rest assured that their documents are safeguarded against unauthorized modifications, providing an unprecedented level of security and peace of mind. Whether it's critical business contracts, personal records, or sensitive legal documents, **TrustDocs** ensures the integrity and authenticity of every document, fostering trust and transparency in the digital realm.

Chapter 3: Flow Diagram



Chapter 4: Sequence Diagram



Chapter 5: Technology Stack

The development of Trust Docs, a blockchain project that verifies documents efficiently and securely, has utilized a robust and well-rounded technology stack. This stack primarily comprises React, Node.js, Express.js, and Truffle Ganache for blockchain development. React, known for its high-performance user interfaces, provides a responsive and user-friendly platform for document submission and verification. While Node.js and Express.js have enabled the building of a scalable, reliable, and lightweight server to handle the core logic of the application, ensuring seamless communication between the user interface and the blockchain network. Additionally, Truffle Ganache has been crucial in facilitating blockchain development by offering a local blockchain environment for testing and development. This local blockchain environment enhances the overall development process by allowing for rapid iteration and smart contract deployment. Together, these technologies have enabled Trust Docs to deliver a secure, efficient, and user-friendly solution for document verification

React

- A JavaScript library for building user interfaces
- Creates high-performance, responsive, and reusable components
- Utilizes a declarative programming paradigm

Node.js

- A JavaScript runtime environment
- Enables the development of scalable and efficient server-side applications
- Popularly used for building web APIs, microservices, and real-time applications

Express.js

- A minimal and flexible Node.js web application framework
- Provides routing, middleware, and template rendering capabilities
- Widely used for building RESTful APIs and web services

Ganache

- A local Ethereum blockchain development environment
- Simulates a real Ethereum blockchain on your local machine
- Provides a convenient platform for testing and deploying smart contracts

MongoDB

- A NoSQL document-oriented database
- Offers flexible data schema and scalability
- Popularly used for storing and retrieving large volumes of data

Chapter 6: Code

```
import React, { useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import { addDocument, checkValidity, getDocuments } from './api';
import {
  Box,
  Text,
  Heading,
  VStack,
  Grid,
  Button,
  useToast,
} from '@chakra-ui/react';
import Navbar from '../components/Navbar.tsx';
import DocumentCard from '../components/DocumentCard';

function Dashboard() {
  const toast = useToast();
  const navigate = useNavigate();
  const [documents, setDocuments] = React.useState(null);
  const [selectedFile, setSelectedFile] = React.useState(null);

  const handleFileInputChange = e => {
    const file = e.target.files[0];
    setSelectedFile(file);
  };

  const addToBlockchain = async () => {
    if (!selectedFile) {
      toast({
        title: 'Select a file',
        description: 'File not selected',
        status: 'error',
        duration: 2000,
        isClosable: true,
      });
      return;
    }

    try {
      const buffer = await selectedFile.arrayBuffer();
      const hashBuffer = await crypto.subtle.digest('SHA-256', buffer);
      const hashArray = Array.from(new Uint8Array(hashBuffer));
      const hashHex = hashArray
        .map(byte => byte.toString(16).padStart(2, '0'))
        .join("");
      const fileName = selectedFile.name;
      console.log(fileName);
      const response = await addDocument(hashHex, fileName);
```



```

if (response === true) {
  toast({
    title: 'Document Added',
    description: 'This document has been added to your blockchain.',
    status: 'success',
    duration: 2000,
    isClosable: true,
  });
} else {
  toast({
    title: 'Something went wrong!',
    description: 'This document could not be added to your blockchain.',
    status: 'error',
    duration: 2000,
    isClosable: true,
  });
}
} catch (error) {
  console.log(error);
}
};

```

```

const calculateFileHash = async () => {
  if (!selectedFile) {
    toast({
      title: 'Select a file',
      description: 'File not selected',
      status: 'error',
      duration: 2000,
      isClosable: true,
    });
    return;
  }

```

```

  try {
    const buffer = await selectedFile.arrayBuffer();
    const hashBuffer = await crypto.subtle.digest('SHA-256', buffer);
    const hashArray = Array.from(new Uint8Array(hashBuffer));
    const hashHex = hashArray
      .map(byte => byte.toString(16).padStart(2, '0'))
      .join("");

```

```

    // Check if a document with the same hash exists
    const documentExists = await checkValidity(hashHex);
    if (documentExists) {
      toast({
        title: 'Document Is Verified',
        description: 'A document with the same hash already exists.',
        status: 'success',
        duration: 2000,

```

```

        isClosable: true,
      });
    } else {
      toast({
        title: 'Cannot verify document',
        description: 'This document does not exist on the blockchain',
        status: 'error',
        duration: 2000,
        isClosable: true,
      });
    }
  } catch (error) {
    toast({
      title: 'Cannot calculate hash',
      description: 'An error occurred when calculating the hash.',
      status: 'error',
      duration: 2000,
      isClosable: true,
    });
  }
};

```

```

useEffect(() => {
  let token = localStorage.getItem('token');
  if (!token) {
    navigate('/login');
  }

  //get the user's documents
  getDocuments().then(res => {
    if (res.data !== null && res.data.length > 0) {
      console.log(res.data);
      setDocuments(res.data);
    }
  });
}, []);

```

```

return (
  <>
    <Box fontSize="xl">
      <Grid p={3}>
        <Box>
          <Navbar />
        </Box>
        <VStack
          spacing={2}
          marginTop={10}
          paddingX={10}
          //align to the start
          align="flex-start"

```

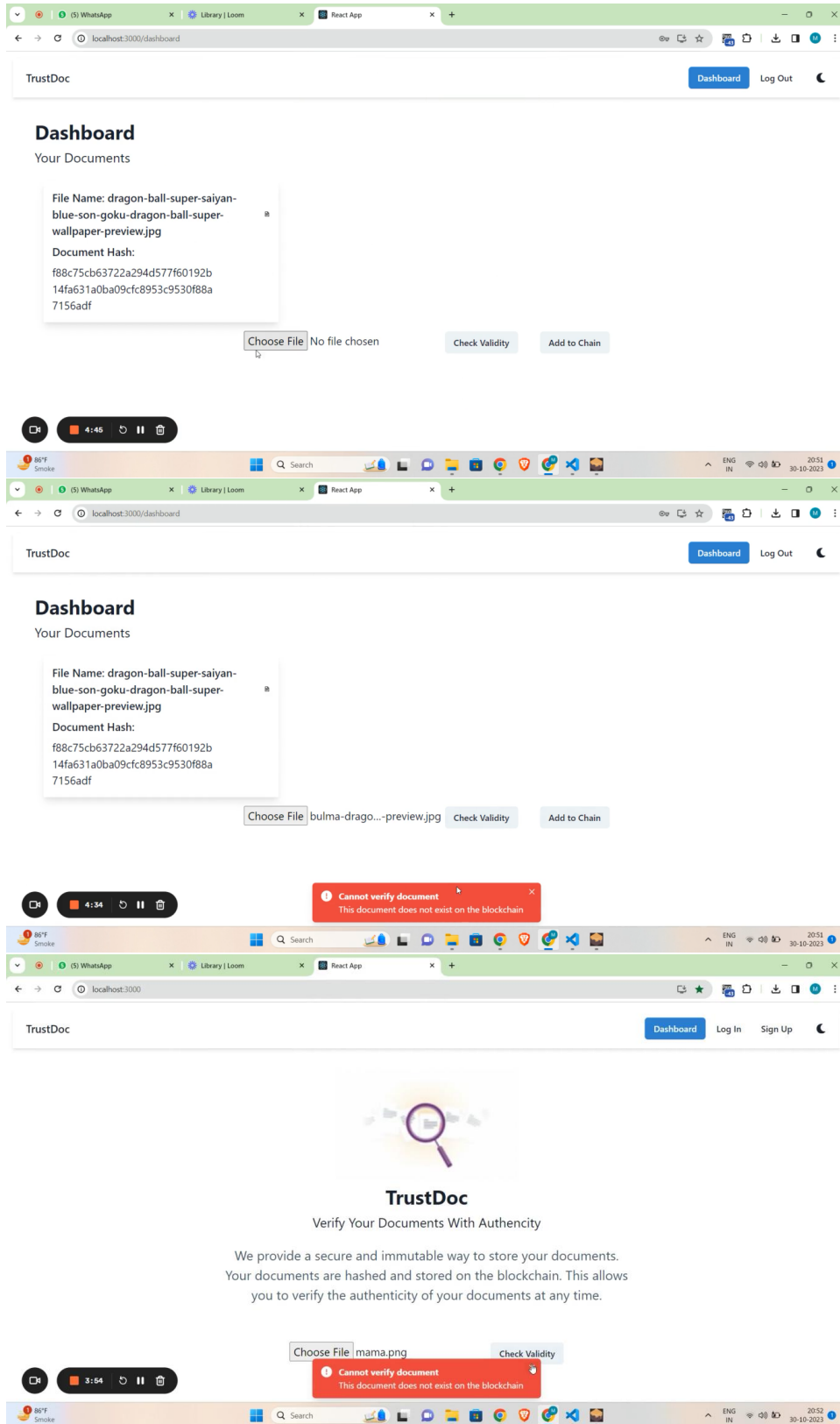
```

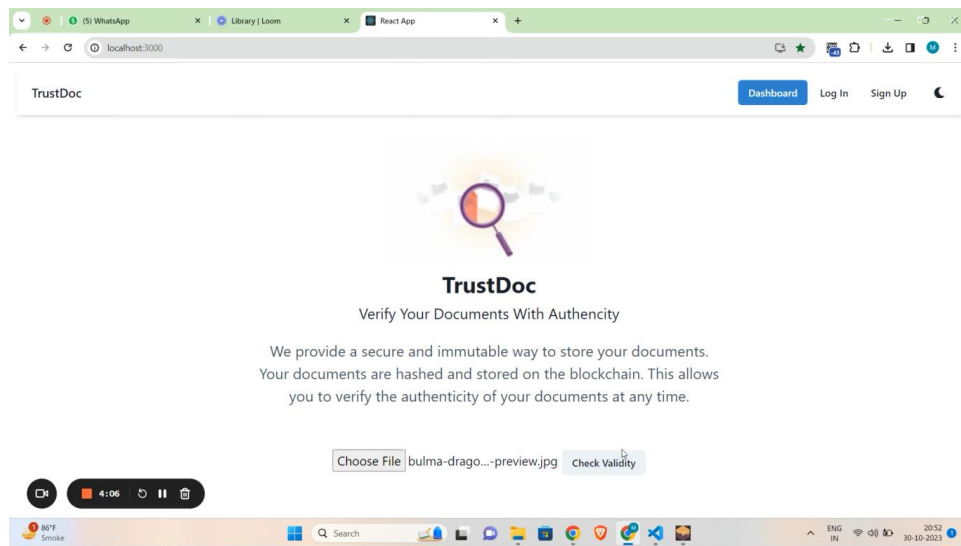
>
<Heading>Dashboard</Heading>
<Text fontSize={'2xl'} fontWeight={'2xl'}>
  {documents === null ? 'You have no documents' : 'Your Documents'}
</Text>
<Grid templateColumns="repeat(3, 1fr)" gap={6}>
  {documents !== null &&
    documents.map(document => <DocumentCard document={document} />)}
</Grid>
</VStack>
<Box align="center">
  <input
    type="file"
    accept="*.*)"
    onChange={e => handleFileInputChange(e)}
  />
  <Button marginRight={10} onClick={e => calculateFileHash(e)}>
    Check Validity
  </Button>
  <Button onClick={e => addToBlockchain()}>Add to Chain</Button>
</Box>
</Grid>
</Box>
</>
);
}

export default Dashboard;

```

Chapter 7: Output





Chapter 8: Conclusion

In conclusion, Trust Docs stands as a beacon of innovation, revolutionizing the document verification landscape with the transformative power of blockchain technology. It decisively addresses the shortcomings of traditional methods and provides an unparalleled user experience. The project's expansive scope extends to individuals, businesses, and organizations, promising to redefine the very essence of document verification. We are immensely excited by Trust Docs' potential to usher in a new era of trust and security. We invite your questions and eagerly anticipate engaging in further discussions. The fully developed product will serve a wide range of users, from individuals to large enterprises, empowering them to navigate the digital world with confidence and peace of mind.