National College of Ireland

Configuration Manual

MSc Research Project
MSc Cyber Security

Sri Satya Prasad Adusumilli
Student ID: 22246754

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sri Satya Prasad Adusumilli

Student ID: 22246754

Programme: MSc Cyber Security **Year:** 2023-24

Module: MSc Research Practicum Part 2

Lecturer: Michael Prior

Submission Due Date: 12-08-2024

Project Title: Blockchain-Based Digital Identity Verification Using Ethereum

Word Count: 1215 **Page Count: 11**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sri Satya Prasad Adusumilli

Date: 12-08-2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sri Satya Prasad Adusumilli
22246754

1 Introduction

This is a configuration guide to the Web3Identity project that seeks to help in the introduction of lending and borrowing that is decentralized through the use of blockchain technology. Contained in this document are the steps, strategies, and processes on how to develop, configure and manage the Web3Identity project. The main purpose of this project is the usage of blockchain technology for the lending and borrowing process, which will be transparent and safe.

In terms of structure, this manual is subdivided into a series of chapters that deal with particular aspects of the project: requirements, installation, configuration and, problems and solutions. The Web3Identity system can be deployed and managed very easily when a user follows all the steps guidelines mentioned in this guide.

2 System Requirements

Hardware Requirements:

- **Minimum CPU:** Intel Core i5 or equivalent
- **RAM:** 8GB
- **Storage:** 100GB SSD
- **Network:** Ethernet or Wi-Fi

Software Requirements:

- **Operating System:** Windows 10, macOS, or Linux
- **Java JDK:** Version 11 or higher
- **Python:** Version 3.8 or higher
- **Docker:** Version 20.10 or higher
- **Node.js:** Version 14 or higher
- **Yarn:** Version 1.22 or higher

Dependencies:

- **Client Dependencies:**
  - React.js
  - Redux
  - Axios
- **Server Dependencies:**
  - Express.js
  - Mongoose
  - JWT
- **Blockchain Dependencies:**
  - Truffle
  - Ganache CLI

- Web3.js

3 Installation Instructions

Step 1: Download and Install Dependencies

1. **Install Java JDK.**
   ```
   sudo apt-get install openjdk-11-jdk
   ```

2. **Install Python.**
   ```
   sudo apt-get install python3.8
   ```

3. **Install Docker.**
   ```
   sudo apt-get install docker-ce docker-ce-cli containerd.io
   ```

4. **Install Node.js and Yarn.**

```
sudo apt-get install nodejs
sudo apt-get install yarn
```

Step 2: Clone Repository

```
git clone https://github.com/your-repository/web3identity.git
cd web3identity
```

Step 3: Install Project Dependencies

Client:

```
cd dapp
yarn install
```

Server:

```
cd backend
yarn install
```

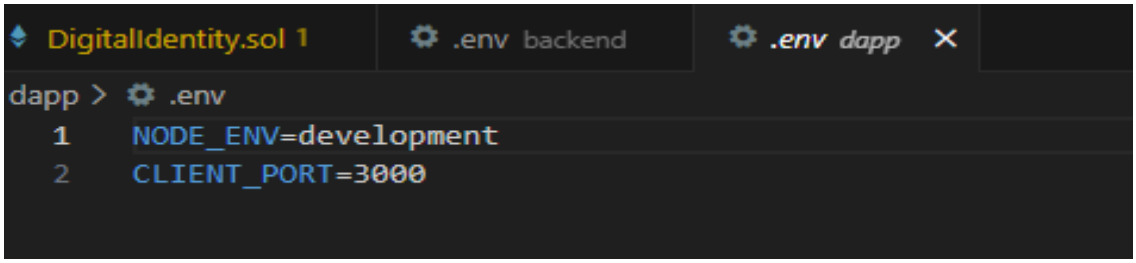
Blockchain:

```
npm install -g truffle
 npm install -g ganache-cli
```

4 Configuration

Client Environment Configuration:

1. Navigate to the dapp directory.
2. Create a .env file with the following content:



Server Environment Configuration:

1. Navigate to the backend directory.
2. Create a .env file with the following content:

Blockchain Environment Configuration:

1. Ensure Ganache is installed and running.
2. Create a truffle-config.js file with the following content:

```
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",
      port: 7545,
      network_id: "*", // Match any network id
    },
  },
  compilers: {
    solc: {
      version: "0.8.0", // Fetch exact version from
solc-bin (default: truffle's version)
    },
  },
};
```

Generate JWT Auth Secret:

Generate a secure random string for the JWT auth secret:

```
node -e
"console.log(require('crypto').randomBytes(32).toStr
ing('hex'))"
```

5 Running the Application

Docker Deployment

1. **Run Docker Compose**:

```
docker-compose up
```

This command will start the Docker containers defined in the docker-compose.yml file.

2. **Build Docker Images**:

```
docker build -t web3identity-server . -f ./deploy/backend.Dockerfile
```

```
docker build -t web3identity-client . -f ./deploy/dapp.Dockerfile
```

These commands build the Docker images for the server and client applications.

3. **Run Containers**:

```
docker run --rm -p 5000:5000 --name web3identity-server-cntr web3identity-server

docker run --rm -p 3000:3000 -p 8545:8545 --name web3identity-client-cntr web3identity-client
```

These commands run the Docker containers for the server and client applications.

Local Deployment

1. **Client**:

```
cd dapp

yarn start
```

This command navigates to the dapp directory and starts the client application.

Server:

```
cd backend

yarn dev
```

This command navigates to the backend directory and starts the server application.

```
PS D:\Satya Blockchain Project U\backend> yarn dev
yarn run v1.22.22
$ nodemon ./bin/www
[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./bin/www`
  backend:db Successfuly connected to MongoDB +0ms
  backend:server listening on port 5000 +3ms
  backend:server app code revision 20240706 1 +0ms
```

Blockchain Deployment

1. **Start Ganache**:

```
ganache-cli
```

This command starts the Ganache CLI, which is used for local blockchain development.

Migrate Smart Contracts:

```
truffle migrate
```

This command deploys the smart contracts to the local blockchain.

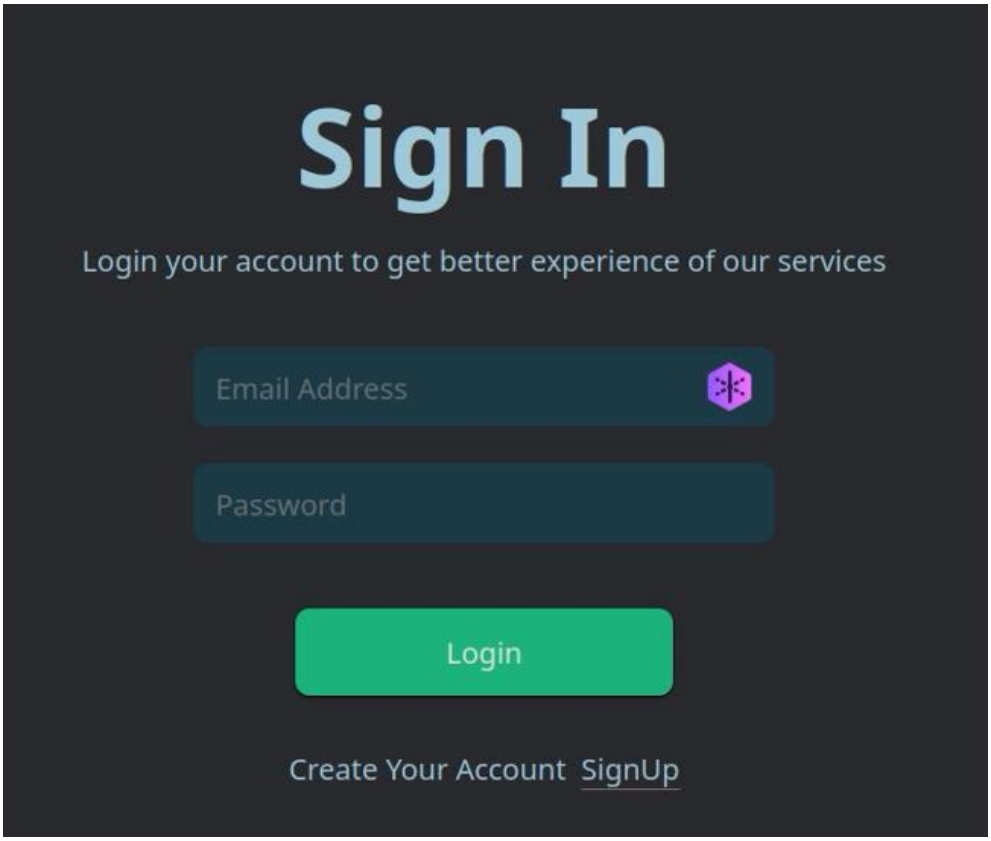
```
PS D:\Satya Blockchain Project U\dapp> truffle migrate

Compiling your contracts...
===========================
> Compiling .\blockchain\contracts\DigitalIdentity.sol
> Compilation warnings encountered:
```

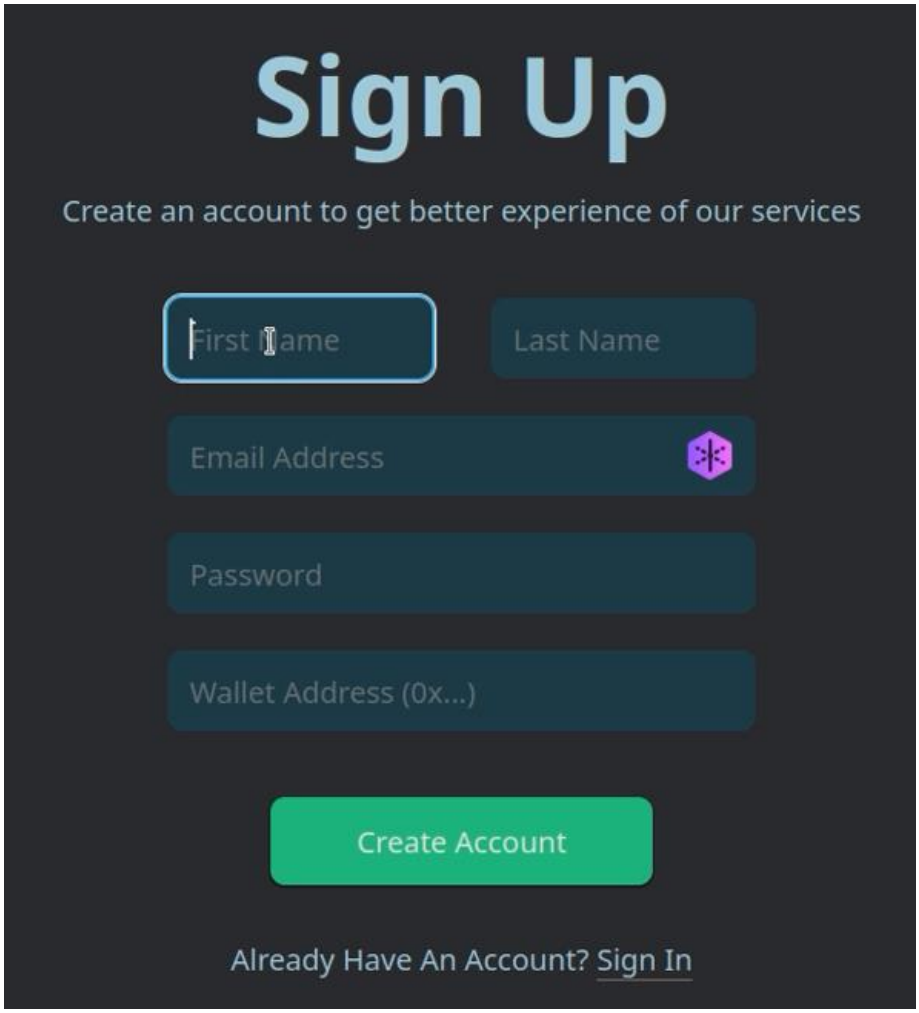
6 API Endpoints

Account Endpoints:

- **Login**
  - Method: POST
  - URL: /account/login
  - Authenticates a user and returns an access token.



- **Signup**
  - Method: POST
  - URL: /account/signup
  - Registers a new user account.



- **Logout**
  - Method: GET
  - URL: /account/logout
  - Ends the user session.

Internal: Users Endpoints:

- **List All Users**
  - Method: GET
  - URL: /users
  - Retrieves a list of all users.

- **Get Single User**
  - Method: GET
  - URL: /users/<id>
  - Retrieves details of a specific user.

- **Create New User**
  - Method: POST
  - URL: /users/create
  - Creates a new user.

- **Delete User**
  - Method: DELETE
  - URL: /users/<id>
  - Deletes a user.

7 Testing and Validation

Run Unit Tests:

1. **Navigate to the Backend Directory**:
   ```
   cd backend
   ```

2. **Run Unit Tests**:
   ```
   yarn test
   ```

Run Integration Tests:

1. **Navigate to the Dapp Directory**:
   ```
   cd dapp
   ```

2. **Run Integration Tests**:
   ```
   yarn test
   ```

8 Troubleshooting

Docker Issues:

- **Docker Daemon Not Running**:
  - Solution: Ensure the Docker daemon is running. Use the following command to start it:

    ```
    sudo systemctl start docker
    ```

- **Docker Logs**:
  - Solution: Check Docker logs for errors using the command:

    ```
    docker logs <container_id>
    ```

Environment Variables:

- **Setting Environment Variables**:
  - Solution: Verify all necessary environment variables are correctly set. Example for setting an environment variable:

    ```
    export VARIABLE_NAME=value
    ```

Network Issues:

- **Network Configuration**:
  - Solution: Ensure proper network configuration and firewall settings.

General Application Issues:

- **Invalid URLs**:
  - Solution: Ensure URLs are correctly formatted and prefixed with http:// or https://.

- **Application Freezes or Crashes**:
  - Solution: Monitor system resources and terminate any processes consuming excessive CPU or memory. Use the top command to monitor resources:

```
top
```

- **Unexpected Output or Incomplete Results**:
  - Solution: Check for errors in the output and ensure all necessary libraries and dependencies are installed correctly.

9 References

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.Available at: https://bitcoin.org/bitcoin.pdf

Buterin, V. (2014). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. Available at: https://ethereum.org/en/whitepaper/

Zyskind, G., Nathan, O., & Pentland, A. (2015). Decentralizing Privacy: Using Blockchain to Protect Personal Data. In 2015 IEEE Security and Privacy Workshops. Available at: https://doi.org/10.1109/SPW.2015.27

Sousa, J. et al. (2018). A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform. In 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).Available at: https://doi.org/10.1109/DSN.2018.00048