A Project Report on

“Web 3 NFT Card Game”

At

“Bhagwan Mahavir College of Computer Application”,

Bharthana-Vesu, Surat

As A Partial Fulfilment for The Degree Of

Bachelor Of Computer Application

2022-23

Guided By:
Asst.Prof. Vipul Chaudhari

Submitted By:
Mr. Hritik R. Jaiswal
Mr. Rishi R. Jaiswal
Mr. Ayush Jain
Mr. Shivam Bansal

Bhagwan Mahavir College of Computer Application

Bharthana-Vesu, Surat

Affiliated With

Bhagwan Mahavir University, Surat,Gujarat,India

BHAGWAN
MAHAVIR
UNIVERSITY

BHAGWAN MAHAVIR COLLEGEOF
COMPUTER APPLICATION

Certificate

This is to certify that the summer project entitled “Web 3 NFT Card Game” has been submitted by **MR.Hritik Jaiswal Exam No. 20020201235** at **Bhagwan Mahavir College of Computer Application**. As a partial fulfilment ofthe requirement for the degree of **Bachelor of Computer Application** for the academic Year 2022-23.

Place: Surat

Date:

Asst.Prof. Himanshi B. Patel
(Project Guide, BMCCA)

Asst.Prof. Hetal R. Modi
(Program Head)

BHAGWAN MAHAVIR UNIVERSITY

BHAGWAN MAHAVIR COLLEGEOF
COMPUTER APPLICATION

Certificate

This is to certify that the summer project entitled “Web 3 NFT Card Game” has been submitted by **<u>MR.Rishi Jaiswal Exam No. 20020201236</u>** at **Bhagwan Mahavir College of Computer Application**. As a partial fulfilment ofthe requirement for the degree of **Bachelor of Computer Application** for the academic Year 2022-23.

Place: Surat

Date:

Asst.Prof. Himanshi B. Patel
(Project Guide, BMCCA)

Asst.Prof. Hetal R. Modi
(Program Head)

BHAGWAN
MAHAVIR
UNIVERSITY

BHAGWAN MAHAVIR COLLEGEOF
COMPUTER APPLICATION

Certificate

This is to certify that the summer project entitled “Web 3 NFT Card Game” has been submitted by **MR.Shivam Bansal Exam No. 20020201** at **Bhagwan Mahavir College of Computer Application**. As a partial fulfilment ofthe requirement for the degree of **Bachelor of Computer Application** for the academic Year 2022-23.

Place: Surat

Date:

Asst.Prof. Himanshi B. Patel
(Project Guide, BMCCA)

Asst.Prof. Hetal R. Modi
(Program Head)

BHAGWAN MAHAVIR UNIVERSITY

BHAGWAN MAHAVIR COLLEGEOF
COMPUTER APPLICATION

Certificate

This is to certify that the summer project entitled “Web 3 NFT Card Game” has been submitted by **Ms.Ayush Jain Exam No. 20020201234** at **Bhagwan Mahavir College of Computer Application**. As a partial fulfilment ofthe requirement for the degree of **Bachelor of Computer Application** for the academic Year 2022-23.

Place: Surat

Date:

Asst.Prof. Himanshi B. Patel
(Project Guide, BMCCA)

Asst.Prof. Hetal R. Modi
(Program Head)

PROJECT PROGRESS REPORT

1. **Project Group No:**
2. **Project Title: Web 3 NFT Card Game**
3. **Project Guide Name: Asst.Prof. Vipul K. Chaudhari**
4. **Progress Report No: 1**
5. **Progress Report Duration:** From 02/01/2023 to 20/01/2023
6. **Project Group Student Names:**

1)Hritik R. Jaiswal	2) Rishi R. Jaiswal
3)Ayush A. Jain	4) Shivam R. Bansal

7. **Project Detail:**
   1) Project Name: Web 3 NFT Card Game (Avax Battlebox)
   2) Front End: React Js (Vite)
   3) Back End: Ethereum Blockchain

8. **Task Completed**

Sr. No.	Student Name(s)	Description
1.	Hritik R. Jaiswal	Visit different website
2.	Rishi R. Jaiswal	Gathered information regarding Project
3.	Shivam B. Bansal	Visit different website
4.	Ayush R. Jain	Gathered information regarding Project

9. **Next Progress Report Date:** 24/02/2023

Project Guide Name:Asst.Prof. Himanshi B.Patel

Project Guide Sign:

Date: -24/02/2023

PROJECT PROGRESS REPORT

1. **Project Group No:**
2. **Project Title: Web 3 NFT Card Game**
3. **Project Guide Name: Asst.Prof. Vipul K. Chaudhari**
4. Web 3 NFT Card Game (Avax Battlebox)**Progress Report No: 2**
5. **Progress Report Duration:** From 21/01/2023 to 24/02/2023
6. **Project Group Student Names:**

1)Hritik R. Jaiswal	2) Rishi R. Jaiswal
3)Ayush A. Jain	4) Shivam R. Bansal

7. **Project Detail:**
   1) Project Name: Web 3 NFT Card Game (Avax Battlebox)
   2) Front End: React Js (Vite)
   3) Back End: Ethereum Blockchain

8. **Task Completed**

Sr. No.	Student Name(s)	Description
1.	Rishi R. Jaiswal	Work on client application
2.	Shivam R. Bansal	Started documentation
3.	Hritik R. Jaiswal	Work on contracts
4.	Ayush A. Jain	Work on downloaded templets

9. **Next Progress Report Date:** 25/03/2023

Project Guide Name: Asst.Prof. Himanshi B.Patel

Project Guide Sign:

Date: -25/03/2023

PROJECT PROGRESS REPORT

1. **Project Group No:**
2. **Project Title: Web 3 NFT Card Game**
3. **Project Guide Name: Asst.Prof. Vipul K. Chaudhari**
4. **Progress Report No: 3**
5. **Progress Report Duration:** From 25/02/2023 to 24/03/2023
6. **Project Group Student Names:**

1)Hritik R. Jaiswal	2) Rishi R. Jaiswal
3)Ayush A. Jain	4) Shivam R. Bansal

7. **Project Detail:**
   1) Project Name: Web 3 NFT Card Game (Avax Battlebox)
   2) Front End: React Js (Vite)
   3) Back End: Ethereum Blockchain

8. **Task Completed**

Sr. No.	Student Name(s)	Description
1.	Hritik R. Jaiswal	Avalanche Config / Writing Contracts
2.	Rishi R. Jaiswal	User Auth and Login Config
3.	Ayush A. Jain	Landing Page and Battle Config
4.	Shivam R. Bansal	Game Assets and Design

9. **Next Progress Report Date:** 29/04/2023

Project Guide Name: Asst.Prof.Vipul Chaudhari

Project Guide Sign:

Date: - 29/04/2023

PROJECT PROGRESS REPORT

1. **Project Group No: 56**
2. **Project Title: Web 3 NFT Card Game**
3. **Project Guide Name: Asst.Prof. Vipul K. Chaudhari**
4. **Progress Report No: 4**
5. **Progress Report Duration:** From 25/03/2023 to 28/04/2023
6. **Project Group Student Names:**

1)Hritik R. Jaiswal	2) Rishi R. Jaiswal
3)Ayush A. Jain	4) Shivam R. Bansal

7. **Project Detail:**
   1) Project Name: Web 3 NFT Card Game (Avax Battlebox)
   2) Front End: React Js (Vite)
   3) Back End: Ethereum Blockchain
8. **Task Completed**

Sr. No.	Student Name(s)	Description
1.	Hritik R. Jaiswal	Unit Testing
2.	Rishi R. Jaiswal	Function Testing
3.	Shivam R. Bansal	Environmental Testing
4.	Ayush A. Jain	Navigational Testing

9. **Next Progress Report Date:** 08/04/2023

Project Guide Name:Asst.Prof.Vipul Chaudhari

Project Guide Sign:

Acknowledgment:

We would like to express our sincere gratitude to all those who contributed to the successful completion of this project. This project marks the culmination of our academic journey and is an essential component of our graduation requirements.

First and foremost, we would like to thank our project guide, Vipul Chaudhari, for his unwavering support and guidance throughout the project. His expertise, feedback, and encouragement were invaluable in helping us navigate the challenges we faced and ensure the project's success.

We would also like to acknowledge the contributions of our team members. [Sample names: John Doe, Jane Smith, Mark Lee, and Sarah Patel] Each member brought a unique set of skills and expertise to the project, and their hard work and dedication were crucial in achieving our project's objectives.

We would like to express our appreciation to the open-source community for developing and maintaining the tools and frameworks we used in this project, including Vite.js and OpenZeppelin. Without these resources, our project would not have been possible.

Lastly, we would like to thank our families and friends for their unwavering support throughout our academic journey. Their encouragement and motivation were instrumental in our success.

Once again, we extend our heartfelt thanks to all those who contributed to this project's successful completion. We are proud of the work we have accomplished and look forward to applying the skills and knowledge we have gained in our future endeavors.

Abstract:

This project involves the development of an NFT card game on the Avalanche platform using Vite.js and OpenZeppelin. The game is designed to help users learn how to deploy applications to the blockchain and interact with it. Avalanche Fuji C-Chain testnet was used for the development and testing of the game.

The project consists of a front-end developed using Vite.js and a smart contract developed using OpenZeppelin. The smart contract is used to create and manage the NFTs, while the front-end provides a user-friendly interface for playing the game.

The game itself is a turn-based strategy game where players battle against each other using their NFT cards. Each card has its own unique abilities and stats, which can be upgraded by acquiring more NFTs or through in-game actions. The game also features a marketplace where players can buy and sell NFTs, allowing them to further customize their deck and strategy.

The project required extensive knowledge of blockchain technology, smart contract development, and front-end web development. The team faced several challenges throughout the development process, including integrating the smart contract with the front-end and ensuring that the game was scalable and secure.

Overall, the project was a success and achieved its goal of providing users with a hands-on learning experience for deploying and interacting with applications on the blockchain. The game demonstrates the potential for NFTs and blockchain technology in gaming and provides a fun and engaging experience for players.

Index

	Topic	Page No.
01.	**Introduction**	1
	1.1 College Profile 1.2 Project Profile	1 3
02.	**Project Overview**	4
	2.1 Technologies Used	4
	2.2 Feasibility Study	6
	2.3 Performance	
03.	**System Analysis**	8
	3.1 Agile Software	8
04.	**Timeline Chart**	4
05.	**Future Development**	8

	5.1 Hardware and Software Requirement	8
	5.1.1 Software Requirement	8
	5.1.2 Hardware Requirement	8
	5.1.3 Technology to be used	8
	5.2 Server-Side and Client-side Tools	13
06.	**System Design**	20
	6.1 Unified Modeling Language(UML) -> Activity, Usecase, Class, SequenceDiagram **OR** 6.1 DFD	14 9
	6.2 Database Design 7.1.1 Data Dictionary 7.1.2 Database Relationship Diagram	17
	6.3 E-R Diagram	16
	6. 4 User Interface Design (Screen Layout)	22
07.	**System Testing (Any Testing According to Project)**	40
	7.1 Unit Testing	
	7.2 Integration Testing	41
	7.3 System Testing	42
08.	**Conclusion**	43
09.	**Future Development**	8
10.	**References**	44
	10.1 Webography 10.2 Bibliography	

1. Introduction

1.1 College Profile

Bhagwan Mahavir College of Computer Application, Surat a highly commendable private institute, occupies a place amongst the technical institutes of the southern region of Gujarat. It is established in year 2008, financed and run by the Bhagwan Mahavir College Foundation, Surat. It is affiliated to Veer Narmad South Gujarat University.

The college is controlled by the executive board formed from the representatives of the Trust under constitution & by laws of the college. The college campus is located at the open greenfield areas with a beautiful location,situated at New city light Road, Bharthana-Vesu, Surat.

The College has excelled itself in many fields. It has occupied a prominent place in the education sphere of South Gujarat. The building surrounded by green fields and contains spacious classrooms, well documented library, a multipurpose hall, modern computer lab. It has at present around 350 students.

Vision

In 21st century, IT technology has been growing in every area of business & society, while in competitive environment it is our vision to make the students more efficientin the field of Information Technology, which will help students to become more efficient skill oriented professional and ready to grab opportunities in the said field.

Mission

Our mission is to provide learning experience through renowned educational institution which shall expose the students in computerapplication in day to day life.Imparting quality undergraduate education in the arena of Computer Application through well designed curriculum and train students to apply this education for life- long high-qualitycareers and give them competitive benefit in the ever-changing and challenging global work environment of 21st century.

Objectives

The College of Computer Application share University's overall missionof being a center of academic excellence by providing qualitative education and integrating training. The progressive and innovative faculty, through their teaching, scholarship and publication, leadership and service are fully committed to the provision of an educational environment which prepares individuals to become successfulprofessionals. The main objective of the BCA program is to help students meet their long-range needs as maturing professionals as well as their immediate needs upon entering the profession.

1.2 Project Technical Profile

Project Title:	Avax Battle Box
Definition:	This application allows users to interact in an anonymous gaming experience where they battle with each other by putting real tokens or cryptocurrency on the line. The main aim of this application as developer was to learn several ways to interact with the blockchain and able to deploy a website on it.
Developed For:	Bhagwan Mahavir College of computer Application,Vesu,Surat.
Project Guide:	Asst.Prof. Vipul K. Chaudhari
Front End:	React JS and Vite
Back End:	Ethereum Blockcahin
Operating System:	Browser Support

Tools used for ERD & DFD:	draw.io
Submitted By:	Mr.Hritik R. Jaiswal Mr.Rishi R. Jaiswal Mr.Shivam R. Bansal Mr.Ayush A. Jain

2. Project Overview:

The Avax Battle Box is a decentralized gaming platform built on the Avalanche blockchain. It allows users to create and play games using the AVAX cryptocurrency. The platform is built on top of the Avalanche-X blockchain network, which is optimized for high-throughput and low-latency transactions.

The project is open-source and available on GitHub, and anyone can contribute to its development. The project is targeted towards gaming enthusiasts who are interested in exploring the potential of blockchain technology in gaming.

2.1Technologies Used:

Solidity: The smart contract language used to implement the game logic.

Avalanche-X: A blockchain network that provides high-throughput and low-latency transactions.

Truffle Suite: A development environment for building, testing, and deploying smart contracts.

Metamask: A browser extension that allows users to interact with the Avalanche network.

IPFS: A distributed file storage system used to store game assets.

Features and Functionalities:

The Avax Battle Box platform has the following features and functionalities:

Users can create and play games using AVAX tokens.

The platform includes a marketplace where users can buy and sell game assets.

The platform is fully decentralized, and all transactions are recorded on the blockchain.

The platform supports multiplayer games, allowing users to compete against each other in real-time.

Limitations and Potential Areas for Improvement:

The platform is still in the early stages of development, and there may be bugs and issues that need to be resolved.

The user interface could be improved to make it more user-friendly and accessible to a wider audience.

The platform currently only supports games built on the Avalanche-X network, limiting the number of potential users.

There may be scalability issues as the platform becomes more popular and attracts more users.

Overall, the Avax Battle Box is an exciting project that has the potential to revolutionize the gaming industry by leveraging the power of blockchain technology. The project is still in the early stages of development, and there are many potential areas for improvement, but it is clear that the development team is committed to building a robust and user-friendly platform.

Detailed Overview of the software used:

- React
- Ethers
- React model
- Web3Modal
- React parallax tilt
- Autoprefixer
- Tailwind
- Hardhat
- Openzeppelin
- Dotenv

React JS

React is a JavaScript library used for building user interfaces. It is maintained by Facebook and has become one of the most popular libraries for front-end development in recent years. Here are some detailed points about React and its pros and cons:

Pros:

- Component-based architecture: React components allow developers to create reusable UI elements that can be easily managed and maintained.
- Virtual DOM: React uses a virtual DOM, which allows for faster updates and better performance compared to traditional DOM manipulation.
- Declarative syntax: React uses a declarative syntax, which makes the code more predictable and easier to understand.
- Large community: React has a large community of developers and users, which means there are many resources and support available for developers.
- React Native: React can be used to build mobile applications using React Native, which allows for the reuse of code between web and mobile applications.

Cons:

- JSX: React uses JSX, which is a syntax extension to JavaScript. This can be confusing for developers who are not familiar with JSX.
- Steep learning curve: React has a steep learning curve, especially for developers who are new to the library or to front-end development.

- Boilerplate code: React requires a lot of boilerplate code, which can make it more time-consuming to get started with a project.
- Limited functionality: React is a UI library and does not include many features that are commonly found in other front-end frameworks.

In summary, React is a popular library for building user interfaces due to its component-based architecture, virtual DOM, and large community. However, it can have a steep learning curve and require a lot of boilerplate code. Understanding the pros and cons of React can help developers make informed decisions when choosing a front-end library for their projects.

Solidity:

Solidity is a programming language that is used to write smart contracts on blockchain platforms like Ethereum, Avalanche, Binance Smart Chain, and others. Here are some detailed points on Solidity:

- Why use Solidity? Solidity is designed to enable developers to write smart contracts for decentralized applications (dApps) that are deployed on blockchain platforms. It is a high-level language that is easy to learn and use, and it provides developers with a range of features that make it easier to write secure and efficient smart contracts.
- Pros of Solidity:
- Secure: Solidity is designed to help developers write secure smart contracts that are resistant to hacking attempts and other security threats.
- Efficient: Solidity is optimized for blockchain platforms, which means that it can execute complex operations quickly and efficiently.
- Open-source: Solidity is an open-source language, which means that developers can contribute to its development and use it freely without any licensing fees.
- Widely used: Solidity is widely used for developing decentralized applications on blockchain platforms like Ethereum, Avalanche, and Binance Smart Chain, which means that there is a large community of developers and resources available for learning and troubleshooting.
- Cons of Solidity:
- Steep learning curve: Solidity has a steep learning curve, especially for developers who are new to blockchain and smart contract development.
- Lack of documentation: Solidity is a relatively new language, and there is a lack of comprehensive documentation and resources available for learning and troubleshooting.

- Limited functionality: Solidity is designed specifically for writing smart contracts, which means that it may not be the best choice for developing other types of applications.

In summary, Solidity is a programming language that is specifically designed for writing secure and efficient smart contracts on blockchain platforms. While it has a few drawbacks, its benefits make it a popular choice for developers who are building decentralized applications.

Various Node Packages:

Here are some of the Node packages used in this project:

1. "openzeppelin/contracts": This package provides a set of pre-built Solidity contracts that implement common functionality like ERC-20 tokens, access control, and more.
2. "@nomiclabs/hardhat-ethers": This package is used to integrate the Ethers.js library with the Hardhat development environment. Ethers.js is a popular JavaScript library for interacting with Ethereum and other EVM-based blockchains.
3. "dotenv": This package is used to load environment variables from a .env file into the project. This is useful for keeping sensitive information like API keys and private keys out of the codebase and under better control.
4. "@nomicfoundation/hardhat-toolbox": This package provides a set of utilities and helpers for the Hardhat development environment. It includes functionality like contract deployment, testing, and more.
5. "react": This is a popular JavaScript library for building user interfaces. It allows for the creation of reusable UI components and provides a declarative syntax for describing how the UI should look and behave.
6. "web3modal": This package provides a simple way to add web3 connectivity to a DApp. It includes support for multiple wallets like MetaMask, WalletConnect, and more.
7. "tailwindcss": This is a CSS framework that makes it easy to style UI components. It provides a set of pre-built classes for common styles like padding, margin, and text alignment, and allows for easy customization and theming.
8. "react-router-dom": This package provides a set of tools for building single-page applications with React. It includes components like <Router>, <Route>, and <Link> for handling navigation and rendering different UI components based on the current URL.

2.2 Feasibility Study:

Technical Feasibility:

The Avax Battle Box platform has been built using a combination of technologies, including Solidity, Avalanche-X, Truffle Suite, Metamask, and IPFS. The use of these technologies has enabled the development team to create a fully decentralized platform that operates on the Avalanche blockchain network.

The platform's smart contracts have been implemented using Solidity, a programming language designed for developing smart contracts on the Ethereum blockchain. The use of Solidity ensures that the platform's game logic is transparent and can be audited by anyone on the blockchain.

The Avalanche-X network provides high-throughput and low-latency transactions, which are essential for a gaming platform that requires fast and secure transactions. The Truffle Suite is a development environment that simplifies the process of building, testing, and deploying smart contracts. Metamask, a browser extension, allows users to interact with the Avalanche network and the platform's smart contracts.

The InterPlanetary File System (IPFS) is a distributed file storage system used to store game assets. IPFS ensures that game assets are available to all users on the network and are resistant to censorship and centralization.

Overall, the combination of these technologies has enabled the development team to build a technically feasible platform that is fully decentralized and operates on a high-performance blockchain network.

Economic Feasibility:

The Avax Battle Box platform is economically feasible as it allows users to create and play games using AVAX tokens, which have real-world value. The platform's marketplace also provides users with the ability to buy and sell game assets, further increasing the economic potential of the platform.

However, the success of the platform will depend on its ability to attract users and establish a strong user base. Additionally, the volatility of cryptocurrency prices and the regulatory environment surrounding cryptocurrency could impact the economic feasibility of the platform.

Operational Feasibility:

The Avax Battle Box platform is operationally feasible as it has been designed to be user-friendly and accessible to a wide range of users. The platform includes a graphical user interface (GUI) that allows users to easily create and play games.

However, there may be scalability issues as the platform becomes more popular and attracts more users. The platform will need to consider strategies for scaling, such as sharding or layer 2 solutions, to ensure that it can handle a large number of transactions and users.

Additionally, the platform will need to consider security and uptime concerns. As a decentralized platform, the Avax Battle Box platform is susceptible to hacking and other security threats. The platform will need to implement robust security measures and ensure that it has a reliable and resilient infrastructure to maintain uptime and prevent disruptions.

Legal and Ethical Feasibility:

The Avax Battle Box platform is subject to legal and ethical considerations, particularly with regards to the use of cryptocurrency and the potential for illegal activities such as money laundering or terrorism financing.

The platform will need to comply with relevant laws and regulations, including anti-money laundering (AML) and know your customer (KYC) regulations. Additionally, appropriate measures will need to be taken to prevent illegal activities, such as monitoring for suspicious transactions and implementing appropriate user verification processes.

The platform will also need to consider ethical considerations such as user privacy and data protection. As a decentralized platform, the Avax Battle Box platform should prioritize user privacy and ensure that user data is protected and not exploited for profit.

Overall, the Avax Battle Box platform appears to be technically, economically, and operationally feasible, but will need to address scalability, security, and compliance concerns to achieve long-term success.

2.3 About performance:

The performance of the system will be dependent on the network conditions and use of wallets. Recommended crypto wallet for this project is Core provided by avalanche. But for the sake of familiarity we are using the most popular wallet in blockchain world meta-mask which has it’s performance issue on the game. At the server end memory requirements will be that of the server software (Operating System, Database Software, Blockchain Network, etc.) and the space required to store the data. The primary memory (RAM) and secondary memory (Hard Disk Space) requirement of the system at the client end will be the same as that required by the web browser and Operating System.

Some general considerations that may impact the performance of the platform.

Blockchain Network: The performance of the blockchain network, which serves as the backbone of the Avax Battle Box platform, can have a significant impact on the platform's performance. The blockchain network's throughput, latency, and stability can affect the platform's transaction processing speed, cost, and security. As more users join the platform and more transactions are processed, the blockchain network may experience congestion, leading to slower transaction processing times and higher transaction fees.

Game Assets: Game assets, such as images, videos, and sound files, are essential components of the games hosted on the Avax Battle Box platform. These assets are stored on the IPFS network, which provides distributed file storage that ensures that the assets are available to all users. However, if the IPFS network experiences downtime or becomes congested, the platform's performance may be impacted. Slow download times or the unavailability of game assets can reduce user engagement and satisfaction.

User Activity: The level of user activity on the platform, including the number of users and transactions, can also impact performance. As the number of users and transactions increases, the blockchain network and IPFS network may become congested, leading to slower transaction processing times and higher transaction fees. Additionally, the platform's servers may become overloaded, leading to slow response times or server crashes.

3. Software process model :

3.1 Agile Development Methodology:

We used the Agile development methodology for this project. Agile is a software development approach that emphasizes flexibility, collaboration, and iterative development. It involves breaking down the project into smaller, more manageable tasks, and prioritizing them based on their importance to the overall project goals.

The Agile methodology allowed us to adapt to changes quickly and ensure that the project was moving in the right direction. We used Agile boards to track the progress of each task and assigned different team members to each task based on their strengths and skill sets.

One of the key principles of Agile development is breaking down a project into smaller, more manageable components. This involves identifying the various tasks required for the project and then splitting them into smaller, more easily achievable parts. By doing so, the project team can tackle each component one by one, which makes the overall project less daunting and more achievable. This also allows for greater flexibility in the development process, as each component can be worked on independently, allowing the team to pivot as needed.

Agile development is a software development approach that emphasizes flexibility and continuous improvement. The main focus of agile development is to deliver software quickly and respond to change efficiently. In agile development, the project is divided into smaller components, known as sprints, hat are developed in short cycles.

The agile development methodology encourages collaboration between cross-functional teams, including developers, testers, and stakeholders, to ensure that the project is delivered on time and within budget. The team works together to identify potential issues and come up with solutions in real-time, which makes the development process more streamlined and efficient.

4. Timeline chart:

Work Tasks	**Dec**		**Jan**				**Feb**				**March**			
Week	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
1.Requirenment Gathering & Analysis														
1.1 Learn Solidity	■	■												
1.2 Requirement Gathering		■												
1.3 Requirement Analysis		■												
Milestone: Requirements Gathered.														
2.Modeling														
2.1 Identifying proposed project profile.			■											
2.2 Identifying Objectives.			■											
2.3 Scope definition.				■										
2.4 Review.				■										
Milestone: Modelling Completed.														
3.System Design														
3.1 Design system flow.					■	■								
3.2 Database Design.					■	■								
3.4 Player Side Design							■							
3.5 Review.							■							
Milestone: System Design completed														
4.Coding														
4.1 Front-End Coding					■	■	■							
4.2 Smart Contracts Coding								■	■					
Milestone: Coding completed														
5.Testing														
5.1 Unit Testing											■	■		
5.2 Integration Testing												■	■	
5.3 Navigational Testing													■	■
5.4 Functional Testing														■
5.5 Performance Testing														■
Milestone: Testing Complete														
6. Documentation			■		■	■		■	■		■	■		■

5. Future Development:

Multiplayer Game Modes: Currently, the platform only supports single-player games. Adding support for multiplayer games would greatly enhance the user experience and increase the social aspect of the platform.

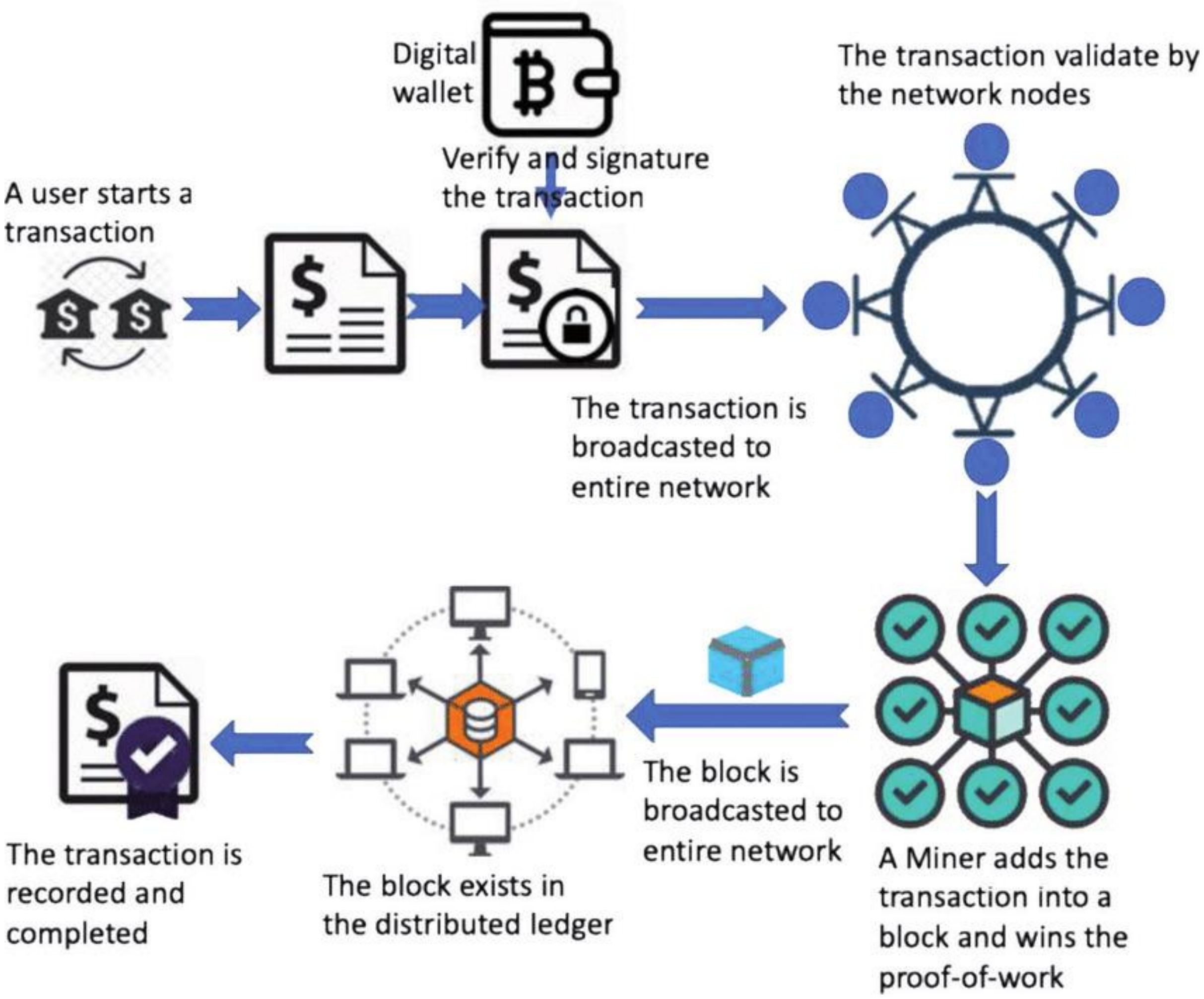
Mobile Application: Creating a mobile application for the platform would greatly increase accessibility and make it easier for users to create and play games on-the-go.

Improved Marketplace Functionality: The platform's marketplace currently allows users to buy and sell game assets. Adding additional functionality, such as auctions or bidding, would increase the economic potential of the platform and enhance the user experience.

Community Features: Adding social features to the platform, such as user profiles, messaging, or forums, would greatly enhance the social aspect of the platform and increase user engagement.

Rewards Program: Creating a rewards program for users who create and play games on the platform would incentivize users to engage with the platform and create high-quality games.

What is Blockchain?


Digital wallet
The transaction validate by the network nodes
Verify and signature the transaction
A user starts a transaction
The transaction is broadcasted to entire network
The transaction is recorded and completed
The block exists in the distributed ledger
The block is broadcasted to entire network
A Miner adds the transaction into a block and wins the proof-of-work

Blockchain technology is a decentralized digital ledger system that records transactions in a secure and transparent way. It consists of a network of computers that work together to validate and verify transactions, with each block of data being linked to the previous block, forming a chain of blocks, hence the name blockchain. This technology has the potential to revolutionize a wide range of industries, including finance, healthcare, logistics, and more.

One of the main features of blockchain technology is its immutability, meaning that once a transaction is recorded on the blockchain, it cannot be altered or deleted. This makes it an ideal solution for industries that require high levels of security and trust, such as finance and healthcare. With blockchain, each transaction is verified by the network of computers, ensuring that there is no single point of failure or corruption.

Another key feature of blockchain technology is its transparency. Since all transactions are recorded on the blockchain, they can be easily accessed and verified by anyone on the network. This creates greater accountability and trust between parties, reducing the risk of fraud and corruption.

In terms of daily life use cases, blockchain technology has the potential to streamline many processes, such as online payments, identity verification, and supply chain management. For example, blockchain-based payment systems can eliminate the need for intermediaries, such as banks, and enable faster, cheaper, and more secure transactions.

In addition, blockchain technology can enable the creation of digital assets, such as NFTs, which are unique digital assets that can be traded and owned by individuals. These assets are created using smart contracts, which are self-executing computer programs that run on the blockchain. The use of NFTs is growing in popularity in the art, gaming, and music industries, allowing for greater ownership and control of digital assets.

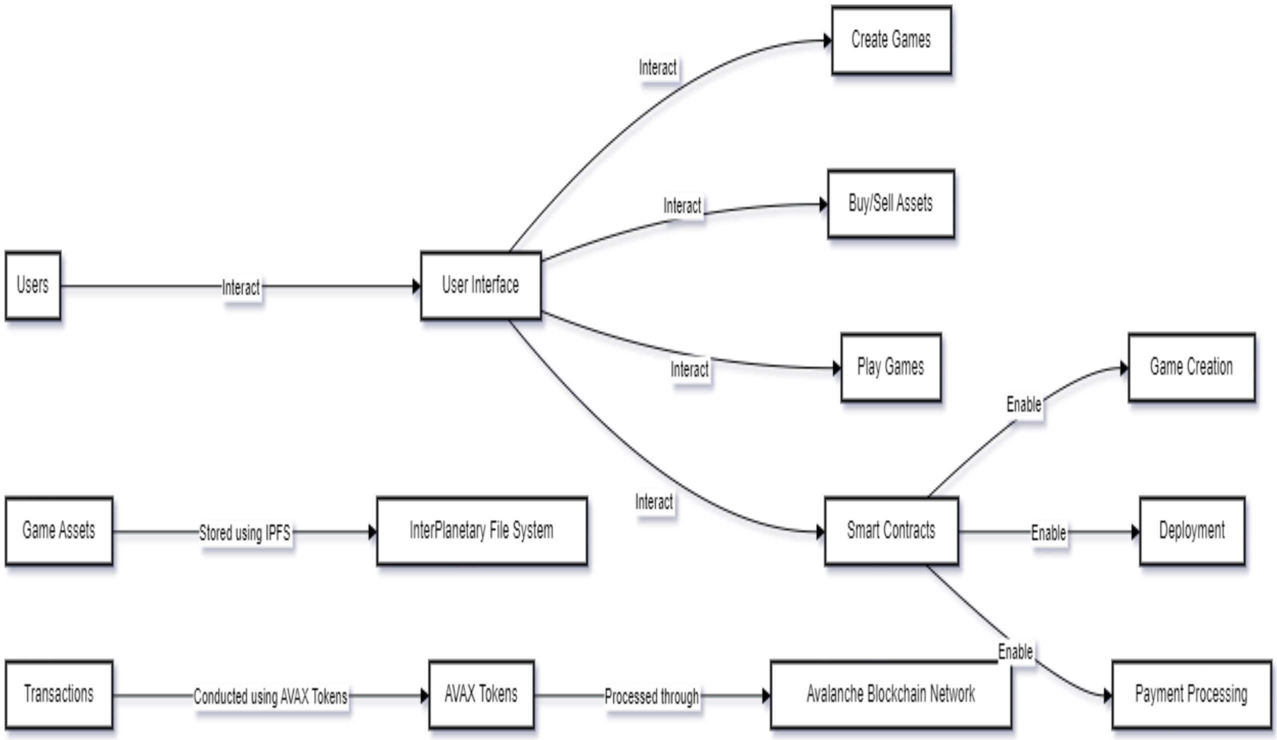
Furthermore, blockchain technology is an essential component of Web 3.0, which is the next generation of the internet. Web 3.0 aims to create a decentralized network and domain system, offering greater security, privacy, and ownership for individuals. This would allow individuals to have more control over their data and online activities, reducing the power of centralized entities, such as social media platforms.

To better understand how blockchain technology works, let's consider a simple diagram. At the core of the blockchain network are nodes, which are individual computers that participate in the network. Each node has a copy of the blockchain ledger, and new transactions are broadcast to all nodes on the network. When a transaction is initiated, it is verified by the nodes on the network, ensuring that it is valid and legitimate. Once the transaction is validated, it is added to a block, which is then added to the blockchain. Each block contains a unique cryptographic code, known as a hash, which ensures the security and immutability of the blockchain.

In conclusion, blockchain technology has the potential to transform many industries, offering greater security, transparency, and ownership for individuals. The use of NFTs and smart contracts is just one example of the many applications of blockchain technology, with new use cases emerging every day. As we move towards a more decentralized and secure internet with Web 3.0, blockchain technology will continue to play a critical role in shaping our digital future.

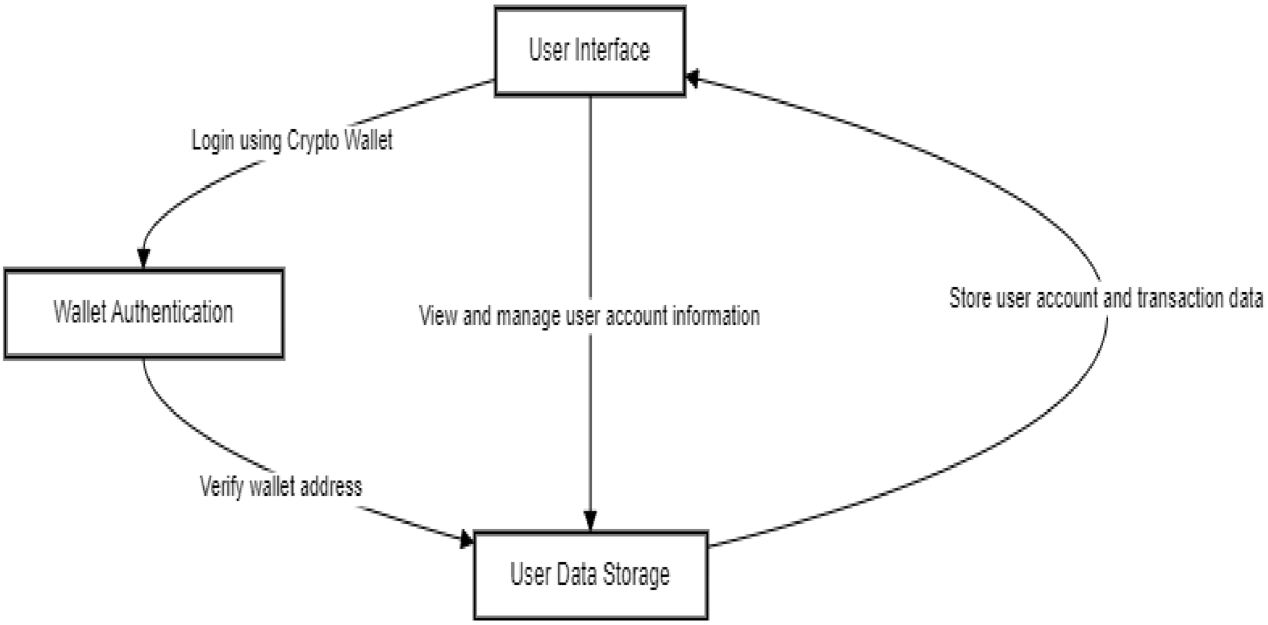
Data Flow Diagram:

Context Level DFD:



The Avax Battle Box platform has two main actors: the users and the Avalanche blockchain network. The users interact with the platform through the user interface, where they can create games, buy/sell assets, and play games. The platform interacts with the Avalanche network through smart contracts, which enable game creation, deployment, and payment processing. Game assets are stored using the InterPlanetary File System (IPFS), and transactions are conducted using AVAX tokens, which are processed through the Avalanche network. Overall, the context level DFD shows the basic interactions between the users, the platform, and the underlying blockchain network, demonstrating how the platform functions to enable users to create, play, and trade games.

User Login DFD:


User Interface
Login using Crypto Wallet
Wallet Authentication
View and manage user account information
Store user account and transaction data
Verify wallet address
User Data Storage

User Interface: The user interface will be the primary way that users interact with the platform. Users will be able to log in to the platform using their crypto wallet, such as Metamask or Core.

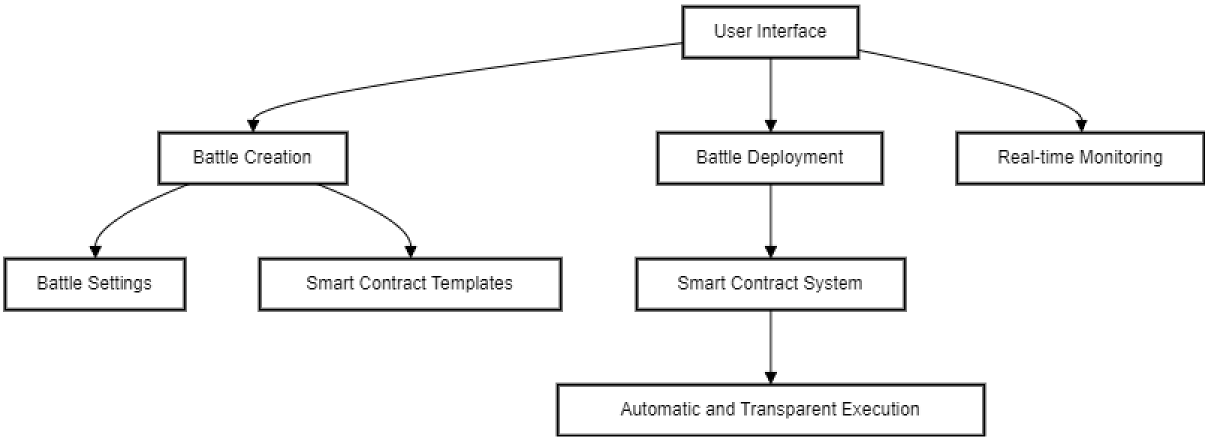
Wallet Authentication: The wallet authentication process will involve verifying the user's crypto wallet address. This will be done using a secure authentication mechanism that is compatible with the user's chosen crypto wallet.

User Data Storage: Once the user has been authenticated, their user data will be stored in a secure database. This will include their crypto wallet address and any other relevant user information, such as their account balance.

User Dashboard: Once the user has logged in, they will be directed to their user dashboard. This will provide access to various platform features, such as creating games, playing games, and buying/selling game assets.

Access Control: Access control mechanisms will ensure that users are only able to access the features and data that they are authorized to access. This will involve interactions between the user dashboard and the secure database.

Battle Creation:



User Interface: The user interface will be the primary way that users interact with the platform. Users will be able to create new battles and join existing battles using the platform's graphical user interface (GUI).

Battle Creation: The battle creation process will allow users to create new battles using the platform's smart contract templates. Users will be able to customize various battle settings, including battle rules, time limits, and asset requirements.

Battle Deployment: Once a battle has been created, it will be deployed to the Avalanche blockchain network. This will require the use of a smart contract deployment tool, such as the Truffle Suite.

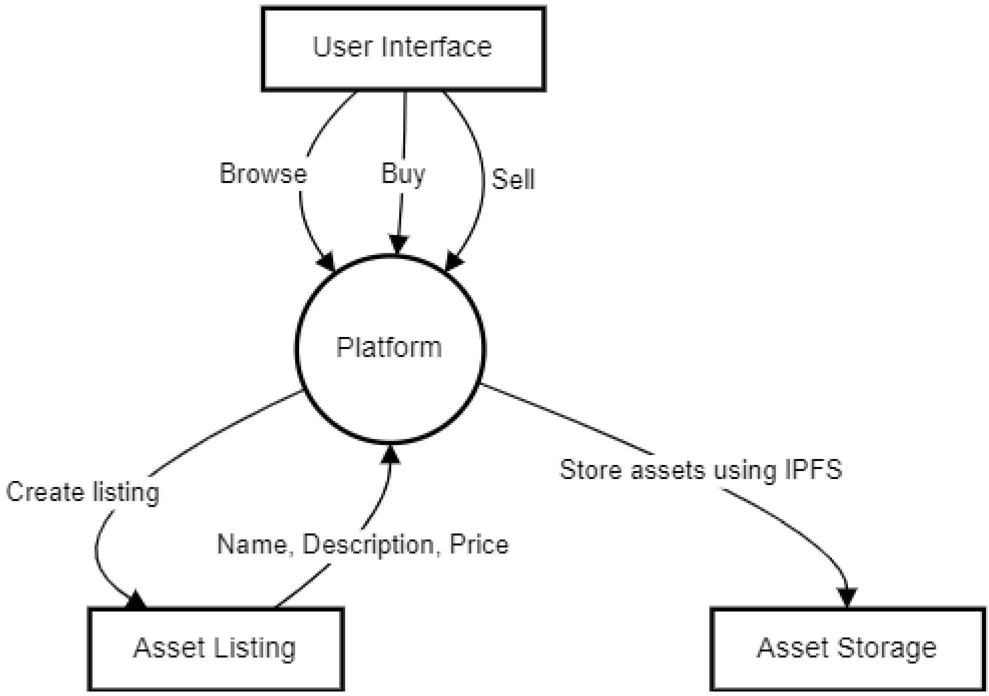
Battle Joining: Users will be able to join existing battles that have been deployed to the blockchain. The battle joining process will involve interactions between the smart contract and the user's game client, which will be connected to the Avalanche network.

Asset Storage: Game assets, such as game characters or backgrounds, will be stored using the InterPlanetary File System (IPFS). This will ensure that game assets are available to all users on the network and are resistant to censorship and centralization.

Asset Requirements: Each battle will have specific asset requirements that must be met by participating users. These requirements will be enforced by the smart contract.

Payment Processing: Payments for battle entry fees and other platform services will be processed using the Avalanche blockchain network. This will involve interactions between the smart contract and the user's digital wallet, such as Metamask.

Trading Platform DFD:



User Interface: The user interface will be the primary way that users interact with the platform. Users will be able to browse, buy, and sell game assets using the platform's graphical user interface (GUI).

Asset Listing: Users who wish to sell game assets will be able to create a listing on the platform. This will involve entering information about the asset, such as its name, description, and price.

Asset Storage: Game assets, such as game characters or backgrounds, will be stored using the InterPlanetary File System (IPFS). This will ensure that game assets are available to all users on the network and are resistant to censorship and centralization.

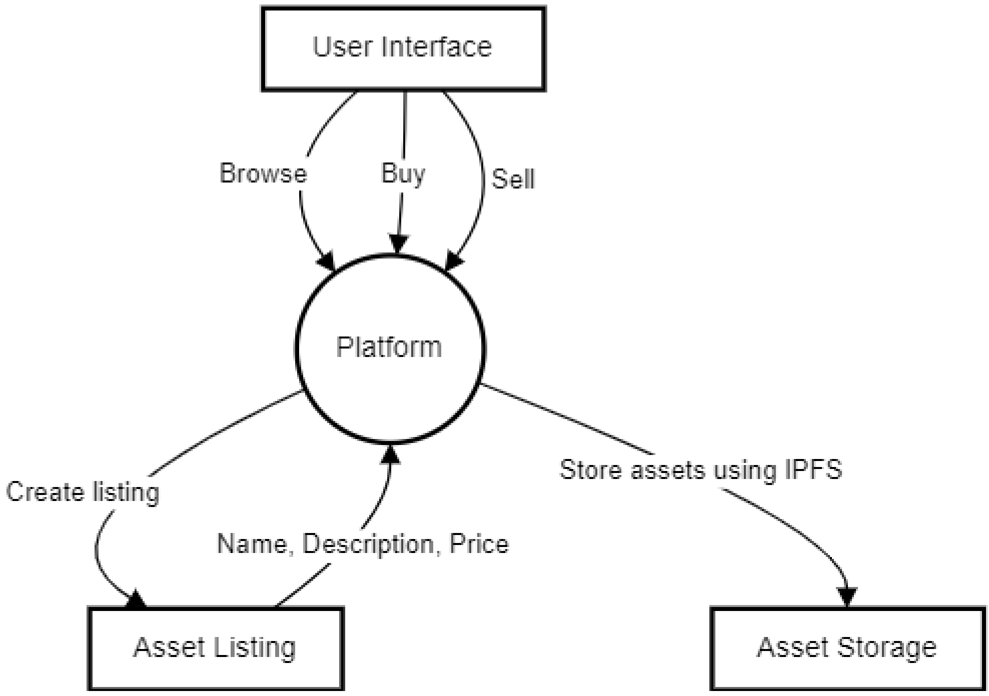
Asset Discovery: Other users will be able to browse available game assets on the platform. This will involve searching for assets based on criteria such as category, price, and popularity.

Asset Purchase: Users will be able to purchase game assets using the Avalanche blockchain network. This will involve interactions between the smart contract and the user's digital wallet, such as Metamask.

Payment Processing: Payments for asset purchases will be processed using the Avalanche blockchain network. This will involve interactions between the smart contract and the user's digital wallet.

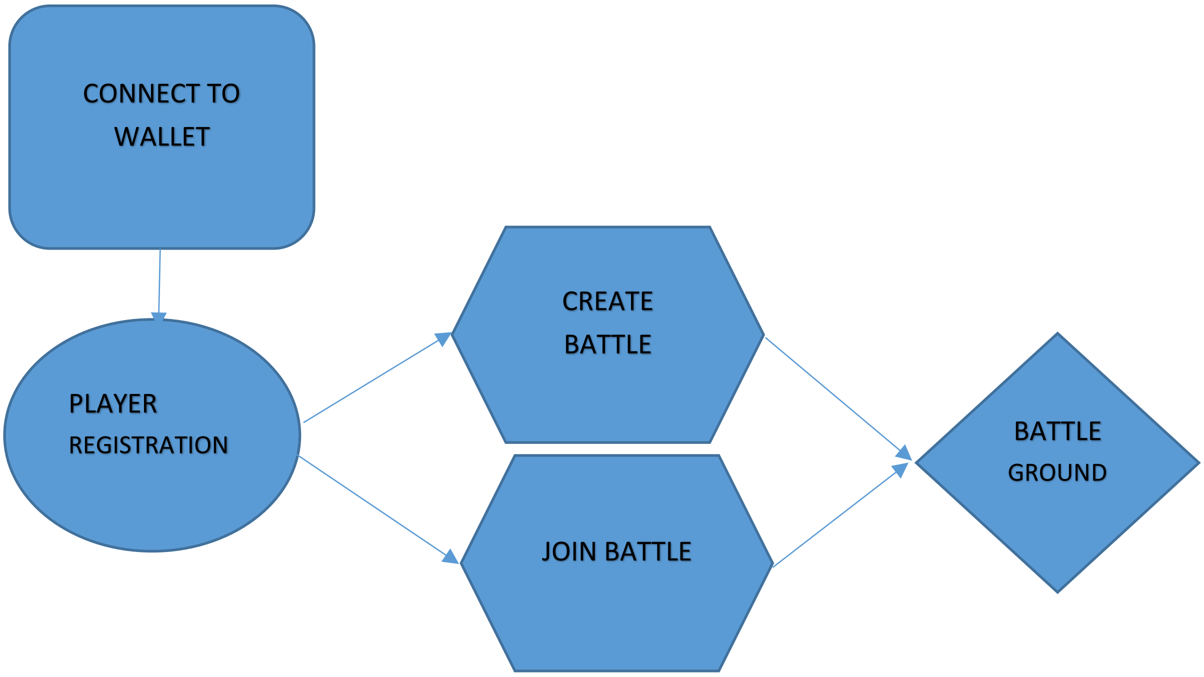
Asset Transfer: Once a user has purchased a game asset, it will be transferred to their digital wallet. This will involve interactions between the smart contract and the user's wallet, as well as the storage and retrieval of the asset from IPFS.

Smart Contract DFD:



Overview of the System:

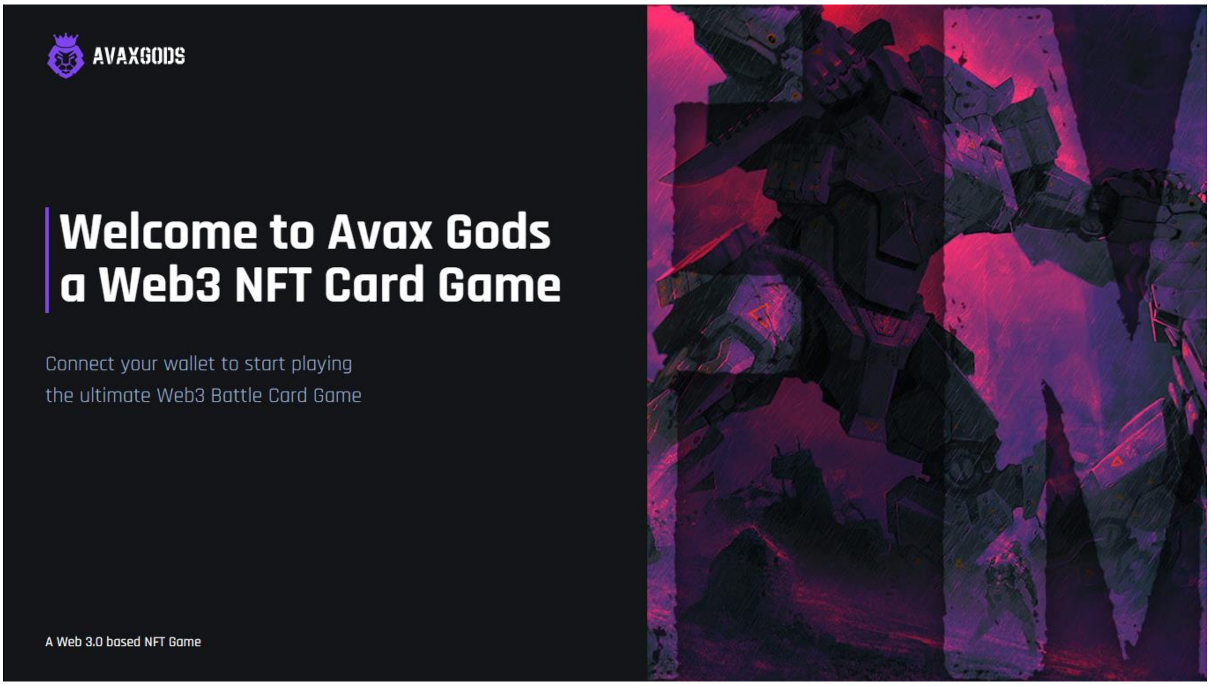
Avax Battle Box is a blockchain-based game built on the Avalanche network. The game is a player versus player (PvP) battle game where players can battle with each other by creating a customized game room. The game has two main components: the frontend and the backend. The frontend is built using ReactJS and the backend uses Solidity to create smart contracts on the Avalanche network. The game allows players to buy and sell game items using cryptocurrency, specifically AVAX.



The above data flow shows a simple flow which would be seen by the user. To give you a more detailed guide to what this whenever a user will first come to this site they will first come to the home page which instructs them to connect the site to a wallet (Whichever they are keen to use) preferably Core wallet. But any of the crypto wallet should be fine, like metamask, WalletConnect, etc. Anyway whenever a user will land on this home immediately they are directed to a pop-up page where they'll be able to connect to the wallet.

Appearance of the application:

Landing page:



AVAXGODS

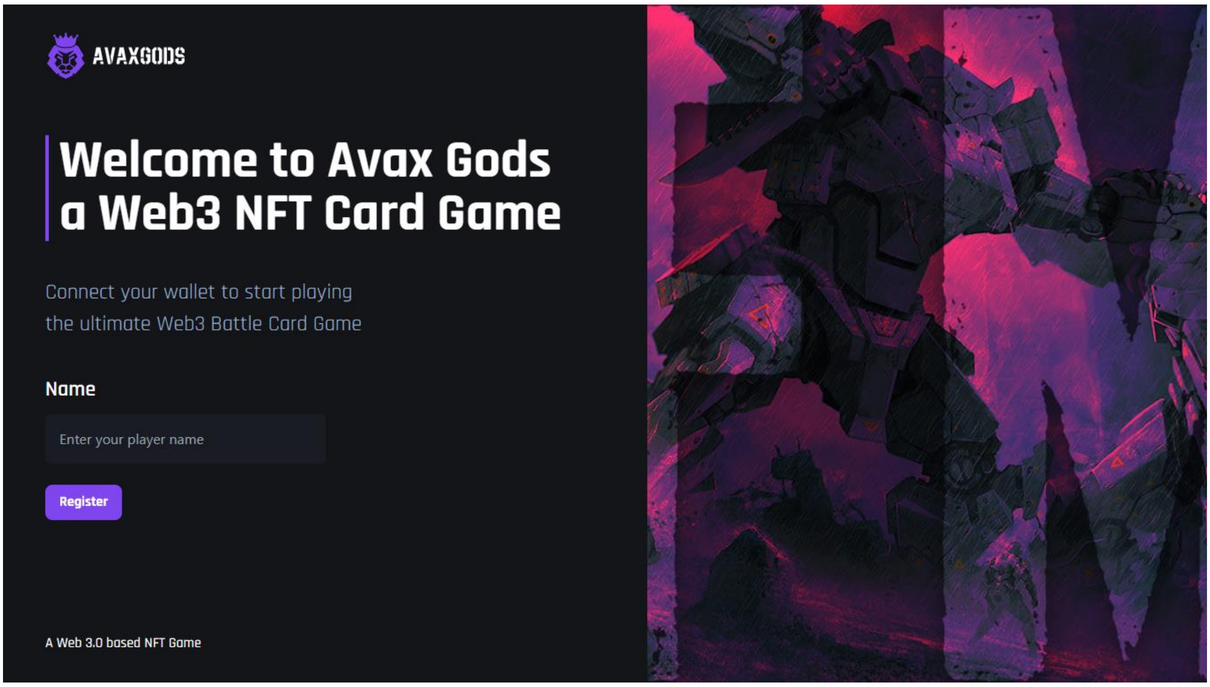
Welcome to Avax Gods a Web3 NFT Card Game

Connect your wallet to start playing the ultimate Web3 Battle Card Game

A Web 3.0 based NFT Game

This is what our landing page looks like. This is the first any first user will see and after connecting to the wallet this page will never reappear to the user. After connecting to the page, then comes the player registration page comes in. It should look like this:-

Registration:



AVAXGODS

Welcome to Avax Gods a Web3 NFT Card Game

Connect your wallet to start playing the ultimate Web3 Battle Card Game

A Web 3.0 based NFT Game

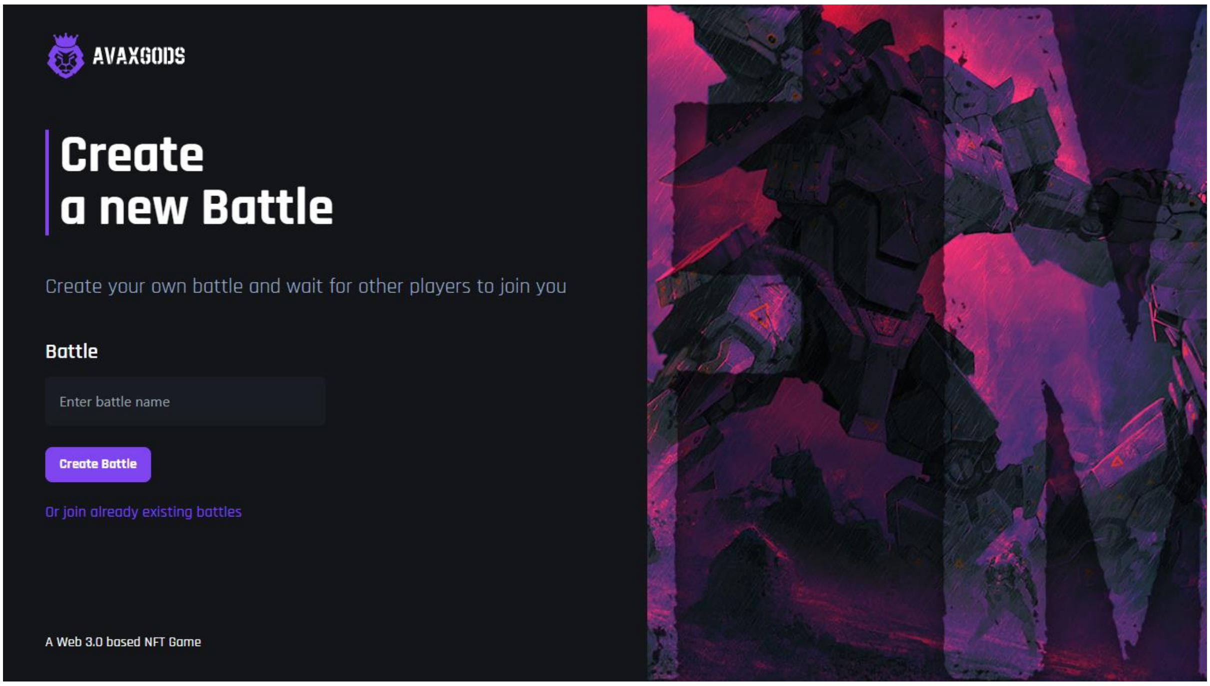
In player registration page users can register their in-game battle names. The name which they'll choose here will be seen to other players whenever they'll battle with others. While registering the name the blockchain will require a small gas fee to integrate the name to its network.

Note that anything which we would do on a blockchain network is immutable. Meaning that anything once entered in a blockchain network will stay like this forever and it can never be altered.

So whenever we are doing something like "Player Registration", Battle Creation", "Attacking" or "Winning or Losing the game" are all recorded on the blockchain. Always there for people to see making it very transparent and accessible with no place for scams and frauds.

After Player registration a token will generate which is like a unique ID for the player and will be used throughout the application to identify the player.

Battle Creation:



AVAXGODS

Create a new Battle

Create your own battle and wait for other players to join you

Battle

Enter battle name

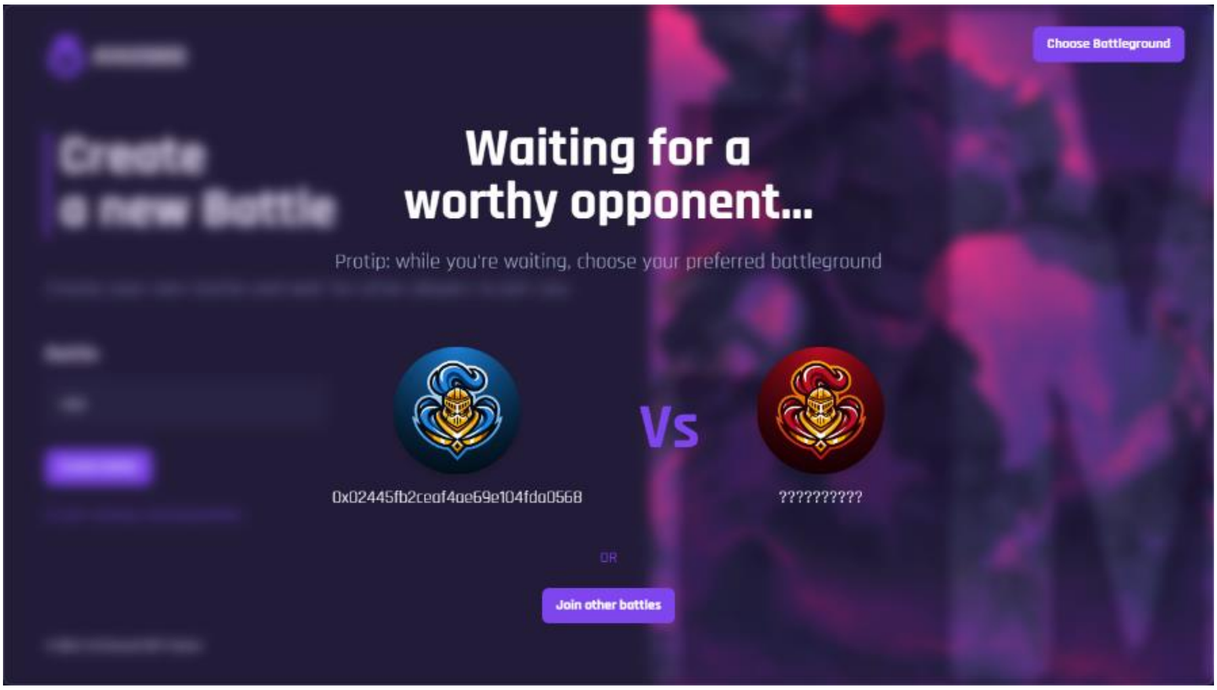
Create Battle

Or join already existing battles

A Web 3.0 based NFT Game

After Player registration, the user is then transferred to battle creation, each battle created by the player will generate a unique ID / token and each player can create only game at a time, its more like that each player can participate only one game at a time.

Waiting screen:



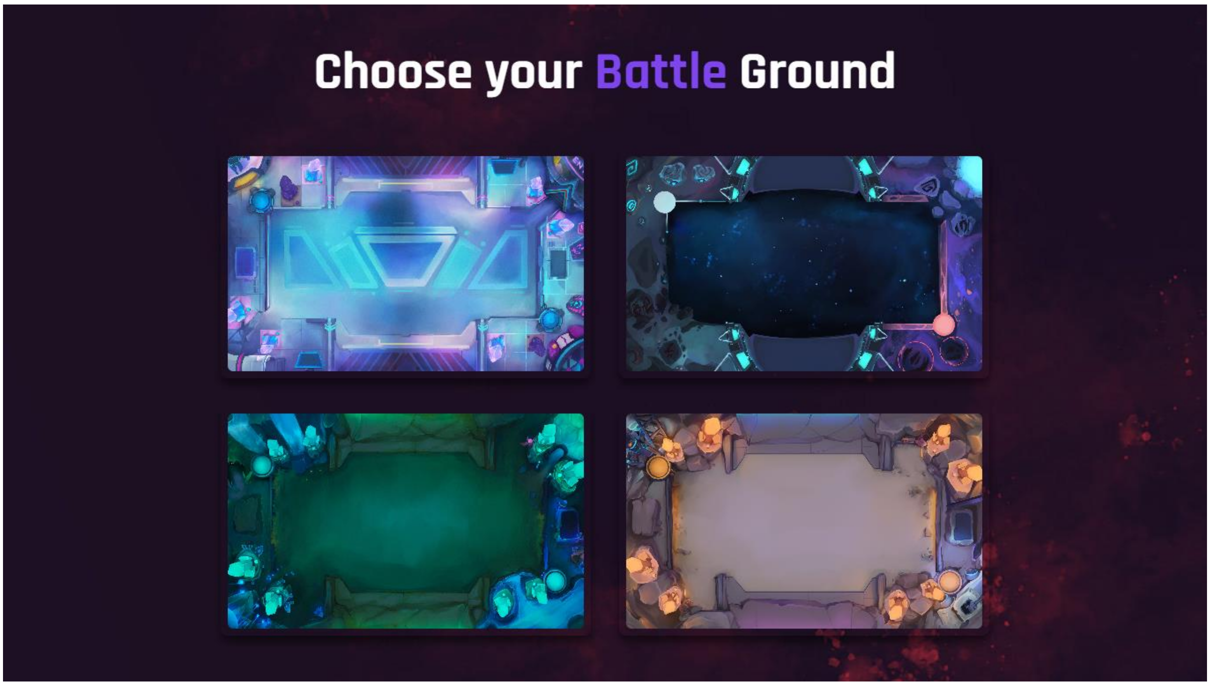
Waiting for a worthy opponent...

Protip: while you're waiting, choose your preferred battleground

This photo on the above is a pop up waiting screen for the player who have created a battle and waiting for other players to join the battle.

The waiting screen consists of leaving the wait area or while waiting choose different battleground look.

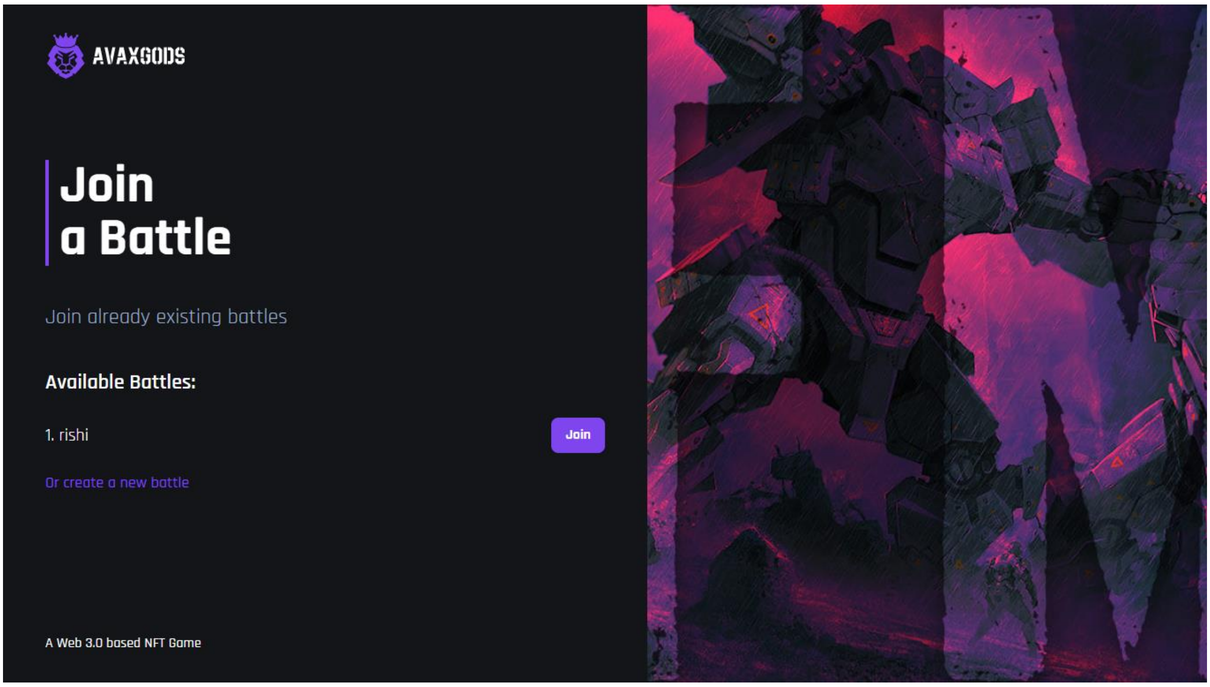
Battle Background:



This are the four battleground look our team has come up with. If required additional cosmetics / new battleground could be added with ease they are after all very easy to make.

Another player could the join the battle by clicking “Join existing battle”, through which the he – she will be transferred to join battle page:

Join Battle:



Join a Battle

Join already existing battles

Available Battles:

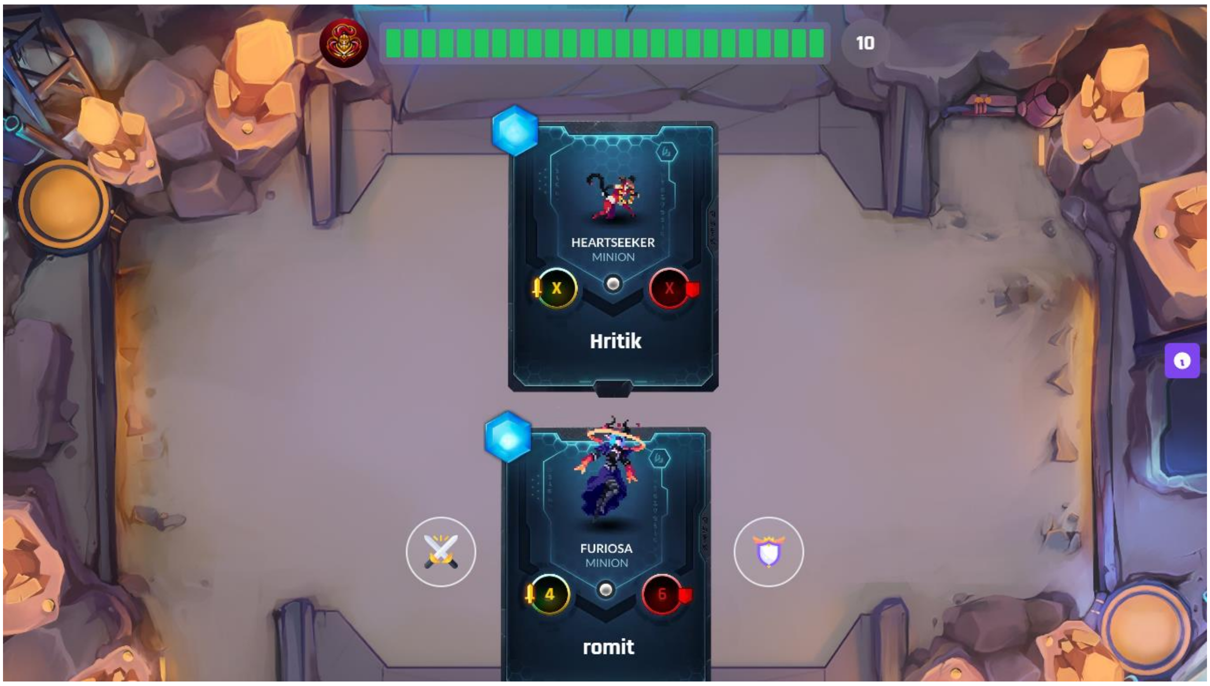
1. rishi

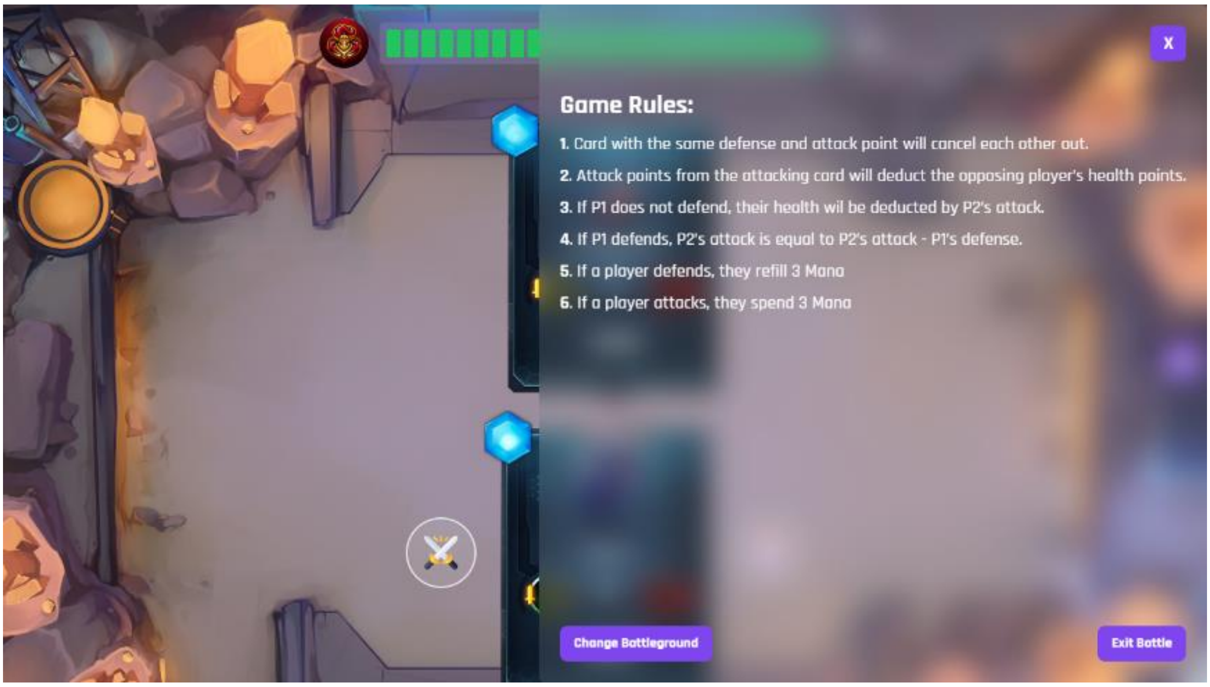
Or create a new battle

A Web 3.0 based NFT Game

Over here on this page players can see ongoing pending battles and can join any battle they wish to join. Each ongoing Pending battle can be seen in available battle section. Joining a battle will create transaction entry in the blockchain confirming that you’ve entered the battle arena.

Batleground Look:





Game Rules:

1. Card with the same defense and attack point will cancel each other out.
2. Attack points from the attacking card will deduct the opposing player's health points.
3. If P1 does not defend, their health wil be deducted by P2's attack.
4. If P1 defends, P2's attack is equal to P2's attack - P1's defense.
5. If a player defends, they refill 3 Mana
6. If a player attacks, they spend 3 Mana

These is what the Battleground should look like. Players can read game rules and can play the game. Winner will take the tokens and losers will lose tokens.

Project Goals:

Provide a platform for users to participate in multiplayer blockchain-based games using their Avalanche wallets: This goal speaks to the primary purpose of the Avax Battle Box platform - to provide users with a space to engage in blockchain-based games using their Avalanche wallets as a means of payment and verification.

Enable users to create and join battles, compete against other players, and earn rewards for their performance: This goal is about creating a compelling gaming experience that motivates users to engage with the platform. By allowing users to create and join battles, compete against other players, and earn rewards for their performance, the platform can incentivize participation and foster a sense of competition among its user base.

Build a user-friendly interface that simplifies the process of creating and joining battles on the Avalanche network: This goal is about ensuring that the platform is accessible to users of all skill levels. By designing a user-friendly interface that simplifies the process of creating and joining battles, the platform can reduce barriers to entry and make it easier for users to engage with the platform.

Implement a secure and scalable infrastructure that can support a large number of simultaneous users and battles: This goal is about ensuring that the platform is robust and can handle a large number of users and battles. By implementing a secure and scalable infrastructure, the platform can ensure that it remains reliable and accessible even as its user base grows.

Foster a vibrant community of players who can connect and interact with each other on the platform: This goal is about building a sense of community around the platform. By fostering a vibrant community of players who can connect and interact with each other on the platform, the platform can create a sense of belonging and increase user engagement.

Implement a secure and scalable infrastructure that can support a large number of simultaneous users and battles: This goal is about ensuring that the platform is robust and can handle a large number of users and battles. By implementing a secure and scalable infrastructure, the platform can ensure that it remains reliable and accessible even as its user base grows.

Technical Architecture:

Front-end Application: The front-end of the Avax Battle Box platform could be developed using modern web development technologies such as ReactJS or VueJS. The front-end would provide the user interface for creating and joining battles, managing user accounts, and displaying battle results.

Web3Modal: To interact with the Avalanche blockchain, the front-end application could use Web3Modal, a JavaScript library that provides a simple and user-friendly API for connecting to wallet providers and interacting with Ethereum-compatible blockchains. Web3Modal can be used to send transactions, read smart contract data, and manage user wallets.

Smart Contracts: The smart contract layer is responsible for managing the business logic of the Avax Battle Box platform. This layer could be developed using Solidity, the most popular programming language for writing smart contracts on the Ethereum platform. The smart contracts would define the rules of engagement for battles, manage user accounts, and handle the transfer of assets.

Avalanche Network: The Avalanche network provides the underlying infrastructure for the Avax Battle Box platform. It is responsible for validating transactions, executing smart contracts, and maintaining the state of the blockchain. The Avalanche network is highly scalable and supports a high number of transactions per second, making it well-suited for gaming and other high-performance applications.

Wallet Providers: The Avax Battle Box platform uses cryptocurrency wallets such as MetaMask, WalletConnect, or Portis, as a means of user authentication and authorization. These wallets store user's private keys and enable them to sign transactions that interact with smart contracts on the Avalanche network. Web3Modal provides an easy way to connect to these wallet providers and manage user authentication.

Server-side components: The Avax Battle Box platform may include server-side components that provide additional functionality such as user authentication, email notifications, or data analytics. These components could be developed using a variety of technologies such as Node.js, Python, or Java.

Testing and Quality Assurance:

Testing and quality assurance processes that have been implemented to ensure that the Avax Battle Box platform meets its requirements and operates as expected.

Unit Testing:

Test Case:	Test Description:	Test Steps:	Result or Errors:
Battle Creation	Verify that a user can create a new battle and that the battle is added to the platform.	Login to the platform using a valid crypto wallet such as Metamask or Core. Click on the "Create Battle" button. Fill in the required details such as battle name, description, and prize. Click on the "Create" button. Expected Result: A new battle is created and added to the platform. The user is redirected to the battle details page.	A new battle is created and added to the platform. The user is redirected to the battle details page.
Battle Joining	Verify that a user can join an existing battle and that their account balance is updated accordingly.	Login to the platform using a valid crypto wallet such as Metamask or Core. Browse the list of available battles and click on the "Join" button for a selected battle. Confirm the transaction and pay the required fee in AVAX.	The user's account balance is updated and they are added to the battle participant list. The user is redirected to the battle details page.
Battle Result Calculation	Verify that the battle results are calculated correctly based on the rules of the game.	Login to the platform using a valid crypto wallet such as Metamask or Core. Join a selected battle and wait for the battle to end. Verify that the battle results are calculated correctly based on the game rules.	The battle results are calculated correctly and displayed on the platform. The winner receives the prize and the battle is closed.
Battle Cancelation	Verify that a user can cancel a battle that they have created.	Click on the "My Battles" option to view the list of battles created by the user. Select the battle that the user wants to cancel and click on the "Cancel" button. Confirm the cancellation and verify that the battle is removed from the platform.	The selected battle is canceled and removed from the platform. Any funds that were collected as a prize are returned to the participants.

Function Testing:

User Registration Function Test Cases:

Verified that the user registration function is able to create a new user account in the system.

Verified that the function is able to detect and prevent duplicate user accounts.

Verified that the user is redirected to the game home page after successfully registering for an account.

Battle Creation Function Test Cases:

Verified that the battle creation function is able to create a new battle with the specified parameters.

Verified that the function is able to detect and prevent duplicate battle names.

Verified that the function is able to set the required number of players and bet amount for the battle.

Battle Joining Function Test Cases:

Verified that the battle joining function is able to add a user to an open battle and deduct the required bet amount from their crypto wallet.

Verified that the function is able to detect and prevent users from joining battles that have already started or are already full.

Battle Result Calculation Function Test Cases:

Verified that the battle result calculation function is able to determine the winner of a battle based on the highest score achieved.

Verified that the function is able to distribute the winner's bet amount among the winning players and return the loser's bet amount to their crypto wallet.

User Profile Function Test Cases:

Verified that the user profile function is able to display the user's crypto wallet address and transaction history.

Verified that the function is able to update the user's profile information such as their display name and profile picture.

Verified that the function is able to delete the user's account and all associated data.

Navigation Testing:

Page Navigation Test Cases:

All page links and buttons are functional and correctly navigate to the expected page.

The browser back button functions as expected and returns the user to the previous page.

The user is redirected to the login page when trying to access a protected page while not logged in.

User Interface Test Cases:

UI elements such as buttons, text fields, and menus are correctly displayed and formatted on all device types.

The UI is intuitive and easy to use, with clear labels and descriptions for all functionality.

The UI is accessible and meets accessibility standards, such as providing alternative text for images and using appropriate color contrast.

Error Handling Test Cases:

Verified that appropriate error messages are displayed when the user enters invalid input or encounters an error.

Error messages are displayed in a clear and understandable manner, and provide guidance on how to correct the error.

Verified that the system is able to recover gracefully from errors and return the user to a functional state.

Security Test Cases:

Sensitive user data such as passwords and crypto wallet addresses are stored securely and encrypted.

Environment Testting:

User Registration Test Cases:

Verify that a user can successfully register for an account using their crypto wallet address.

Verify that the user cannot register for an account with an invalid or already registered crypto wallet address.

Verify that the user receives an email confirmation after successfully registering for an account.

Battle Creation Test Cases:

Verify that a user can create a new battle by specifying the required parameters such as battle name, start date, and end date.

Verify that the user cannot create a battle with an invalid or already used battle name.

Verify that the user can set the required number of players and the bet amount for the battle.

Battle Joining Test Cases:

Verify that a user can join an open battle by paying the required bet amount.

Verify that the user cannot join a battle that has already started or is already full.

Verify that the user's bet amount is deducted from their crypto wallet upon joining the battle.

Battle Result Calculation Test Cases:

The winner of a battle is determined based on the highest score achieved during the battle.

The winner's bet amount is distributed among the winning players as per the rules of the battle.

Verified that the loser's bet amount is returned to their crypto wallet.