



COS30049-COMPUTING TECHNOLOGY INNOVATION PROJECT

Assignment 1 for Group Set 1



CLASS 01
GROUP-1.3

Contents

I.	Project Background and Introduction	2
II.	Team Introduction.....	2
III.	The Value of our NFTs	3
IV.	Project Requirement List and Description	3
V.	Project Design.....	5
1.	Overall System Architecture Design	5
2.	Front-end Prototype	6
3.	Backend Database Design.....	10
	Users table.....	10
	Transactions	10
	Cards table	11
4.	API Design.....	11
	API Name: User Authentication API.....	11
	API Name: User Registration	12
	API Name: Password Reset	13
	API Name: User Balance Update	13
	API Name: Team's performance API	14
	API Name: Blockchain	14
5.	Function Description.....	15
	Functionality Name: User Authentication and Registration	15
	Functionality Name: Digital Asset Listings.....	18
	Functionality Name: Search and Filter	20
	Functionality name: Purchasing NFTS	22
	Functionality name: View Transaction History	24
	Functionality name: Track Club Performance in real-time (Cutting-edge).....	25
	Functionality name: Create an NFT.....	25
6.	Project Deployment Instruction	27
VI.	Conclusion	36
VII.	References.....	36

I. Project Background and Introduction

The Swinball Football Card NFT Trading Platform is a revolutionary marketplace designed to facilitate the buying, selling, and auctioning of digital football player cards as NFTs using blockchain technology. Built on the Ethereum blockchain, our platform ensures the security and immutability of card ownership data, providing users with a transparent and immutable trading experience.

In this project, we aim to revolutionize the football card trading community by utilizing the power of blockchain technology to enhance ownership security and transparency. Traditional physical football card trading is vulnerable to counterfeiting and theft, and traditional digital trading cards are vulnerable to data manipulation and hacking, presenting risks to users' assets and transactions. By implementing a smart contract blockchain system, we provide state-of-the-art ownership security, ensuring that users' football card NFTs are protected from unauthorized access and manipulation.

II. Team Introduction

1. Dinh Duc Minh

- Role: Leader, Project Manager
- Responsibilities: Ensure the project is on track, oversee the project, work on the system architecture design, AWS server deployment and documentation. Ensure the quality of the documents.

2. Tran Viet Thang

- Role: Developer, Quality Assurance
- Responsibilities: Database designer and deployment (Create Index and Query used for further function), search and filter function developer, documentation, ensure the quality of pages, and documents.

3. Doan Minh Hieu

- Role: Developer, Quality Assurance
- Responsibilities: Develop main functions of the website, front-end developer and interface designer. API designer for different functions authentication, resetting password, registration, Buy SwinCoin function. Configure the connection between the front-end and database to run the website's function.

4. Le Nho Bach

- Role: Developer
- Responsibilities: Develop blockchain system and deploy it to AWS, develop team's performance cutting-edge API.

5. Truong Minh Son
 - Role: Developer
 - Responsibilities: Documentation, Video Editing.

III. The Value of our NFTs

In our marketplace, Non-Fungible Tokens (NFTs) hold significant value and continue to attract buyers for several compelling reasons:

- **Uniqueness:** Each NFT represents a one and only digital asset, whether it's a collectible card featuring a football player or an iconic moment captured in digital artwork. The rarity and uniqueness of these digital assets make them highly coveted among collectors and fans alike. To ensure that the website's token IDs are never duplicated, we have implemented the following function to generate the token ID:

This function combines the name, club, and season information to create a unique string. Then, it removes any hyphens from the season string. Finally, it converts the combined string into a hexadecimal format using the 'base64' format. The resulting hexadecimal string serves as the token ID for the website, ensuring uniqueness and preventing duplication.

card_id	image_url	player_name	club	position	age	country	price
QnVrYXlviFNha2EgLSBBcnNlbmFs	bukayo_saka.jpg	Bukayo Saka	Arsenal	Forward	22	England	0.52
Sm9hbyBQYWxoW5oYSAtIEZ1bGhhbSBGQw==	joao_palhinha.jpg	Joao Palhinha	Fulham FC	Midfielder	28	Portugal	0.36
Sm9yZGFuIFBpY2tmb3JkIC0gRXZlcnRvbiBGQw==	jordan_pickford.jpg	Jordan Pickford	Everton FC	Goalkeeper	30	England	0.15
TWF0dHkgQ2FzaCAtIEFzdG9uIFZpbGxhIEZD	matty_cash.jpg	Matty Cash	Aston Villa FC	Defender	26	Poland	0.40

- **Ownership and Authenticity:** NFT ownership is securely recorded on the blockchain, providing proof of ownership and authenticity. Unlike physical football cards susceptible to counterfeiting or loss, NFTs offer security and permanence, instilling confidence in buyers and adding intrinsic value to each token.
- **Investment Potential:** Similar to traditional collectible assets, there's potential for NFTs to appreciate in value over time, particularly if associated with popular players, iconic moments or limited edition releases. Some buyers view NFTs as an investment opportunity, anticipating future increases in value or demand.
- **Community and Fandom:** Football NFTs tap into the passionate and dedicated community of football enthusiasts worldwide. Owning NFTs featuring favorite players or memorable moments allows fans to express their support with their favorite players, fostering a sense of community and emotional connection that adds more value to the tokens.

IV. Project Requirement List and Description

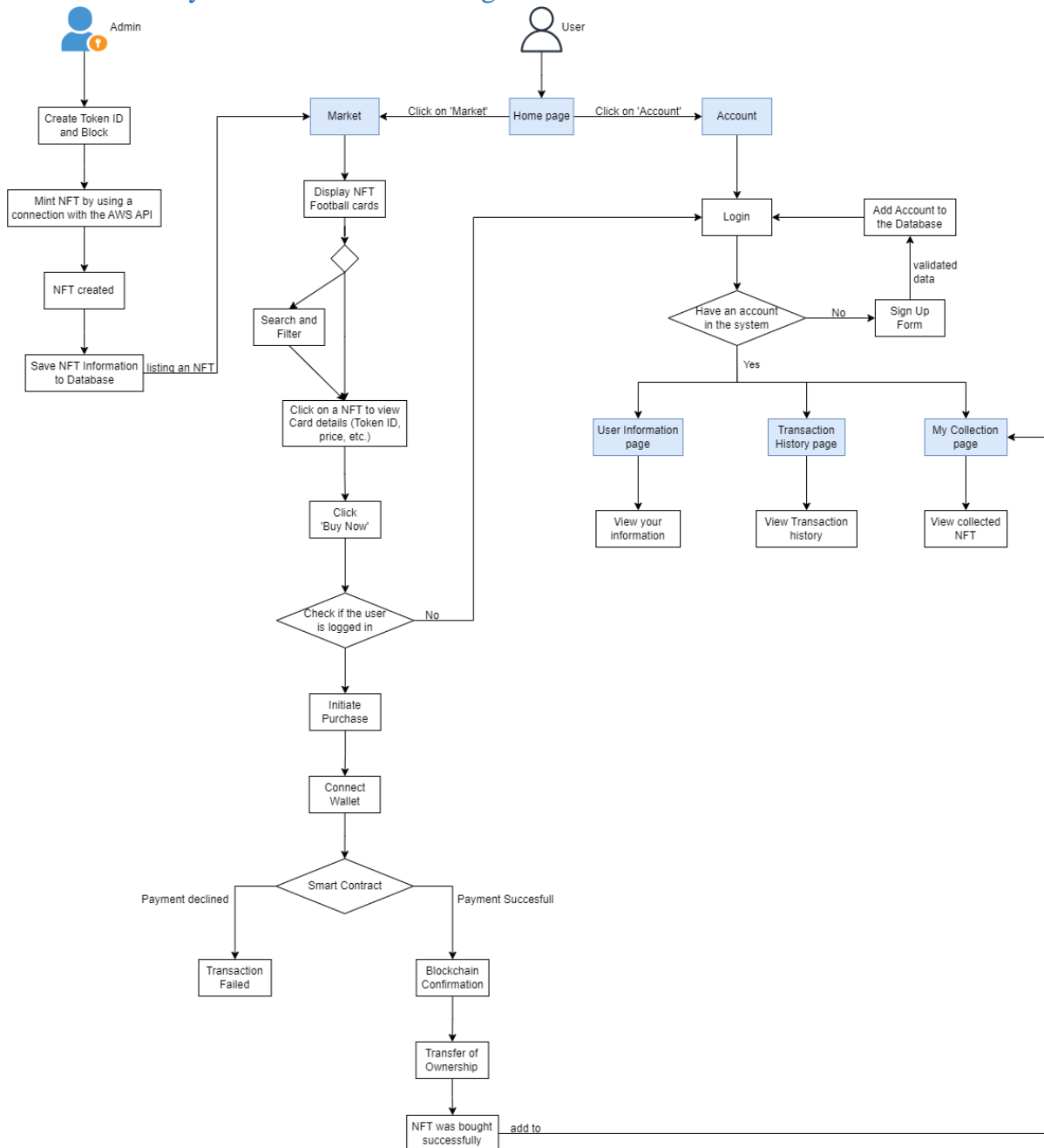
The Project Requirement List outlines the essential features and functionalities required for the successful development of a project including:

Assignment 2 COS30049 – Group 1.3

- Acquire expertise in back-end web development to construct dynamic and data-centric web applications.
- Assess project requirements and convert them into functional back-end features that harmonize with the user interface.
- Obtain practical skills in integrating advanced technologies relevant to the domain, such as Blockchain and Web 3.0, to facilitate smooth interaction between front-end and back-end elements.
- Engage in productive collaboration within a team setting, with each member actively contributing towards accomplishing the project objectives outlined for the second development phase.

V. Project Design

1. Overall System Architecture Design



This is the workflow of the Swinball Football Card NFT project. The system architecture I've designed is robust, secure, and user-friendly. It includes features such as user authentication and registration, digital asset listings, search and filter functionality, smart contracts for escrow, a trading process, and a transaction history. Users can create accounts, view and sort digital assets, initiate trades, and view their past trades. We use Ethereum and Solidity to implement smart contracts that act as escrows during the trading process, ensuring security and transparency in transactions. This innovative platform turns football lovers favorite player moments into digital

assets, allowing football enthusiasts to connect with their favorite players and moments in a unique way.

2. Front-end Prototype

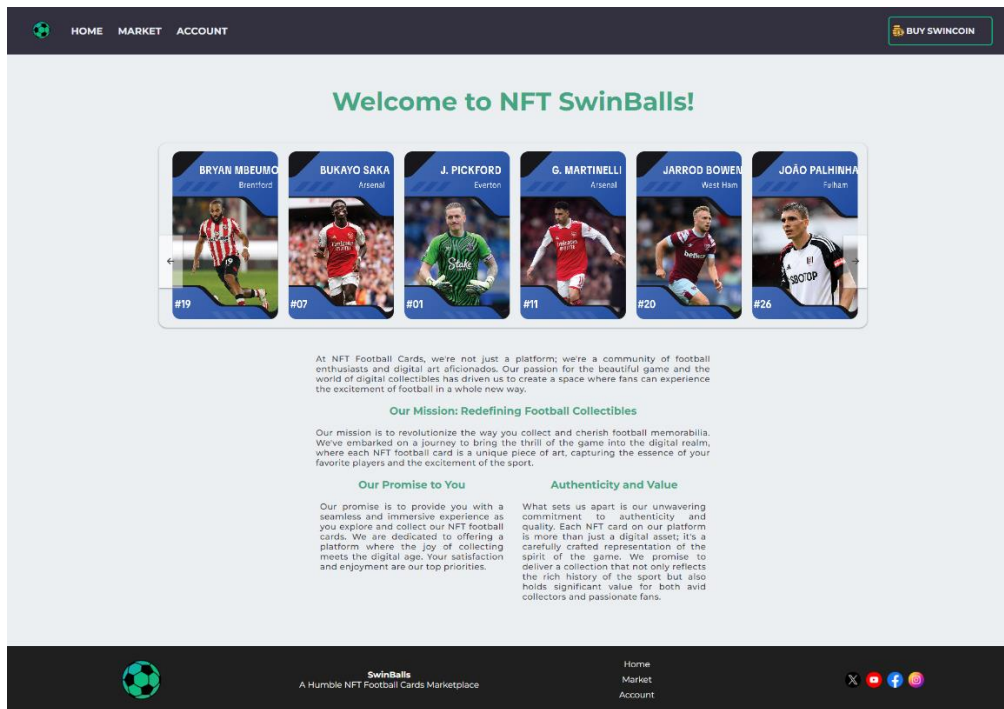
In Assignment 2, our focus was on enhancing the user experience and making the platform more user-friendly compared to the front-end prototype developed in Assignment 1. Here are the key improvements and changes we made:

- Removal of 'Collection' Page: Instead of having a separate 'Collection' page, we decided to integrate a filter function directly into the 'Market' page. This streamlines the user experience by eliminating the need for users to navigate to a separate page to filter and view specific collections. With the filter function integrated seamlessly into the main page, users can easily refine their search criteria and view relevant digital assets without any extra clicks or navigation.
- Addition of Balance on the Navigation Bar: We recognized the importance of providing users with quick access to see their balance. To enhance user convenience, we added the user's balance to the navigation bar. This allows users to see their current balance at a glance, regardless of which page they are on within the platform. Having this information readily available helps users make informed decisions while browsing and trading digital assets without the need to navigate to a separate page to check their balance.

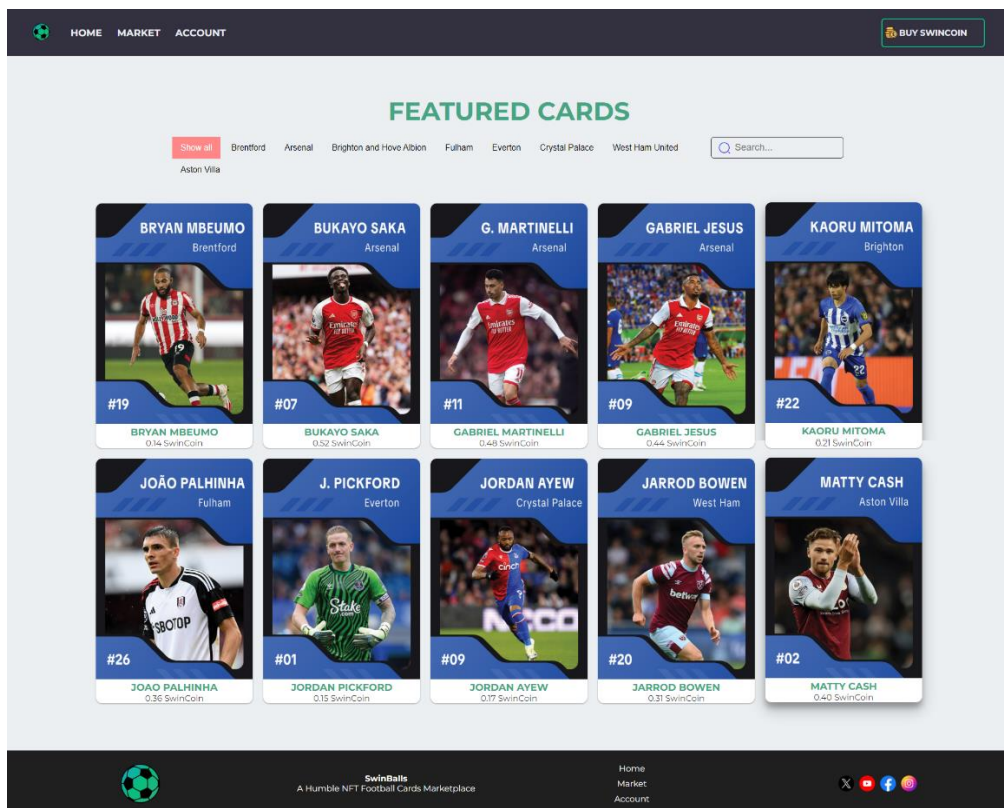
Here is the Front-end of our “Swinball” NFT market place:

Assignment 2 COS30049 – Group 1.3

Home Page:

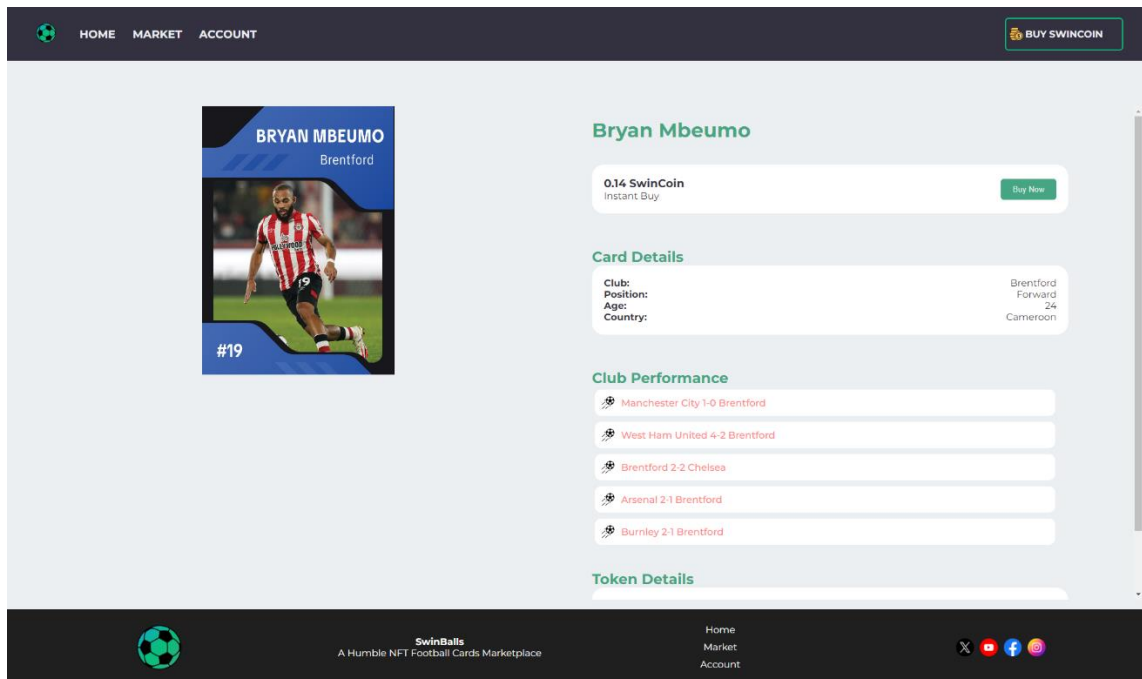


Market Page:

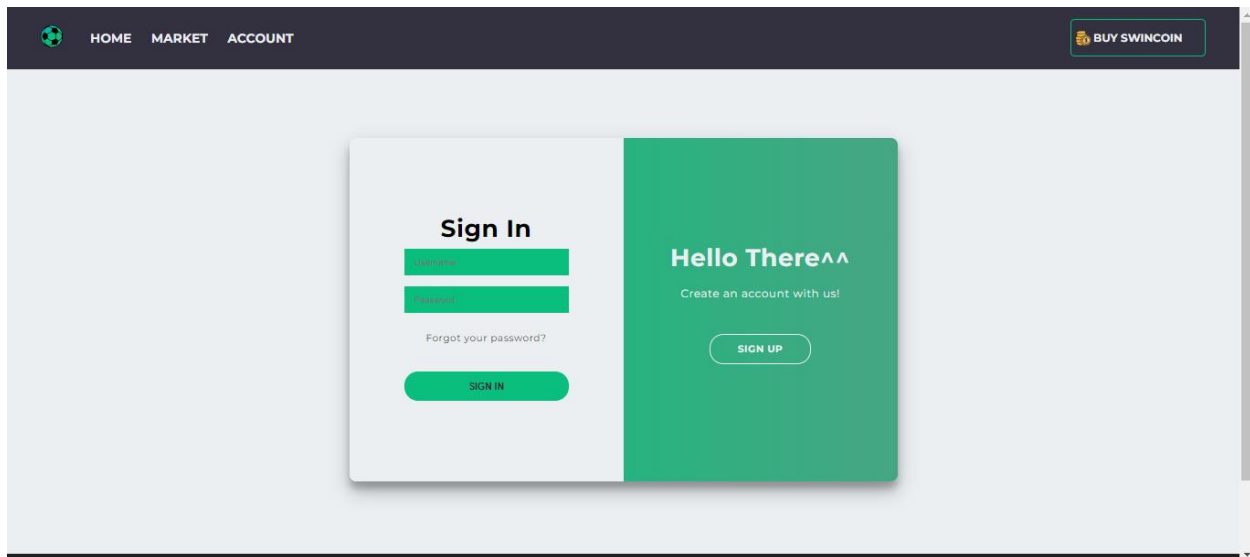


Assignment 2 COS30049 – Group 1.3

Card Details:








Sign In and Sign Up Page:




Assignment 2 COS30049 – Group 1.3

User Profile:

[HOME](#) [MARKET](#) [ACCOUNT](#) [BUY SWINCOIN](#)



**bachle**
ID: 8

[Personal Details](#) [Transactions](#) [Log Out](#) [Reset Password](#)

USER PROFILE


Name : bachle

Email : bachle@aws.com





Date Joined : 2024-03-26

Balance : 1.50 SwinCoin

[Buy 0.5 SwinCoin](#)






**SwinBalls**
A Humble NFT Football Cards Marketplace


[Home](#) [Market](#) [Account](#)

Transaction History:

[HOME](#) [MARKET](#) [ACCOUNT](#) [BUY SWINCOIN](#)




**A Random User**
ID: 123456





[Personal Details](#) [Transactions](#) [Log Out](#) [Reset Password](#)

TRANSACTION HISTORY

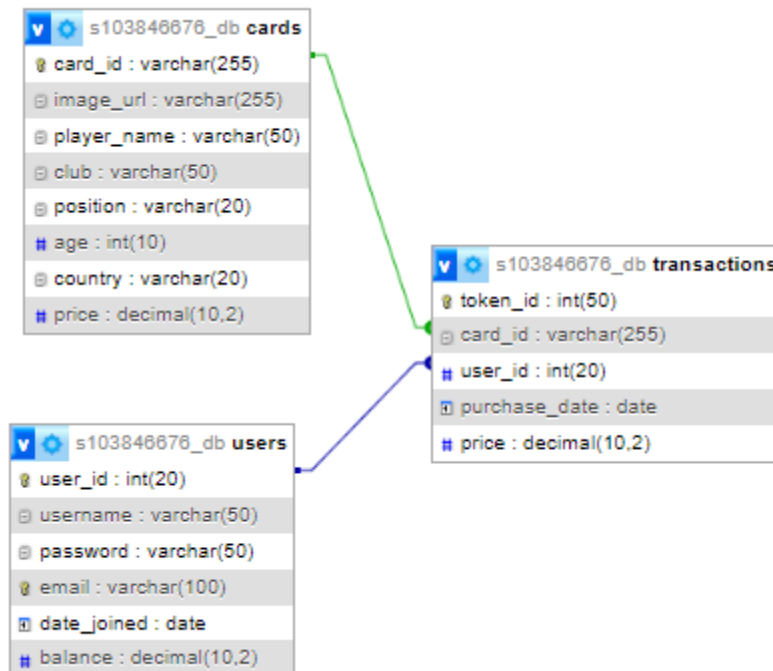
Token ID	Card ID	User ID	Purchase Date	Price
0	QnVYXlviFNhaZEgLSBBcnNlbnRmFs	8	2024-03-25	0.52 SwinCoin

**SwinBalls**
A Humble NFT Football Cards Marketplace

[Home](#) [Market](#) [Account](#)

3. Backend Database Design



The backend database design for the Swinball Football NFT trading platform consists of three main tables: users, transactions, and cards. The first table is the Users table, which stores information about registered users of the platform, the table includes 6 fields, each field stores a personal detail of the user. The second one is the Transactions table, which records the information of the transactions made on the website, which contains 4 fields. The last one is the Cards table which we use to store the information of the cards that the users are seeking, this table has 8 details. These are the detailed information of each table:

Users table

- user_id: Unique identifier for each user, auto incremented for each new user.
- username: User's username, not null.
- password: User's password is securely stored, not null.
- email: User's email address, unique, not null.
- date_joined: Date when the user joined the platform, not null.
- balance: User's available fund for purchasing cards, stored as a decimal, with a precision of 10 and scale of 2.
 - Primary Key: user_id
 - Unique Key: email (to ensure each email address is associated with only one user)

Transactions

- token_id: Unique identifier for each transaction.
- card_id: Foreign key referencing the card_id in the Cards table, indicating the card involved in the transaction.

- `user_id`: Foreign key referencing the `user_id` in the Users table, indicating the user involved in the transaction.
- `purchase_date`: Date when the transaction occurred.
 - Foreign Keys: `card_id` (references `card_id` in Cards table), `user_id` (references `user_id` in Users table)
 - Index: `token_id` column is indexed for improved query performance.

Cards table

- `card_id`: Unique identifier for each card, generate unique ID by converting Player Name + Club to base64.
- `image_url`: URL of the image representing the card.
- `player_name`: Name of the football player featured on the card.
- `club`: Football club to which the player belongs.
- `position`: Position of the player.
- `age`: Age of the player.
- `country`: Country of the player.
- `price`: Price of the card, stored as a decimal, with a precision of 10 and scale of 2.
 - Primary Key: `card_id`
 - Index: `card_id` column is indexed for improved query performance.

This database design provides a solid foundation for the Swinball Football NFT trading platform in storing user information, card data, and transaction records. It allows efficient querying and retrieval of relevant information to support the platform's functionality, such as user information display, user authentication, card listings, and transaction processing.

Since this database is intended for demonstrating our demo website, its primary focus is on showcasing the basic functionality of the Swinball Football NFT trading platform, such as user registration and user information display, card listings, transaction recording, trading. While we understand the benefits of having Index, its impact is somewhat limited in this demo environment because of the modest size of the dataset.

Though we have already implemented Index on the `card_id` of Cards and `transaction_id` of Transactions as these are commonly used in queries involving transactions and card information. However, given the limited scale of data in this demo version, the impact of the Index on the website's performance may not differ. Once this website is published and with larger datasets and heavier traffic, the difference in using Index may be witnessed more clearly, especially in terms of performance and scalability.

4. API Design

API Name: User Authentication API

Description: The User Authentication API provides a service for authenticating users based on their username and password. It verifies the provided credentials against the database and returns a success message if the user is authenticated successfully.

Endpoints:

URL: /authenticate.php

HTTP Method: POST

Request Parameters:

userName2 (string): The username of the user attempting to authenticate.

userPassword2 (string): The password of the user attempting to authenticate.

Response Format: The API responds with JSON format.

Status Codes:

- 200 OK: Successful authentication.
- 400 Bad Request: Missing required username or password.
- 401 Unauthorized: Incorrect username or password.
- 405 Method Not Allowed: Only POST method is allowed.

[API Name: User Registration](#)

Description: The User Registration API allows users to register for a new account by providing their username, email, and password. It checks for duplicate emails in the database and inserts the new user into the database upon successful registration.

Endpoints:

URL: /registration_api.php

HTTP Method: POST

Request Parameters:

- email (string): The email address of the user registering.
- userName (string): The desired username for the new account.
- userPassword (string): The password for the new account.

Response Format: The API responds with JSON format.

Status Codes:

- 200 OK: Successful registration.
- 400 Bad Request: Missing required username, password, or email; or registration cannot be processed.
- 405 Method Not Allowed: Only POST method is allowed.
- 409 Conflict: Email already exists in the database.
- 500 Internal Server Error: An error occurred during the registration process.

API Name: Password Reset

Description: The Password Reset API allows users to reset their password by providing their username and email. It verifies the user's credentials against the database and updates the password if the user exists.

Endpoints:

URL: /resetpassword_api.php

HTTP Method: POST

Request Parameters:

- email3 (string): The email address of the user requesting a password reset.
- userName3 (string): The username of the user requesting a password reset.
- userPassword3 (string): The new password for the user account.

Response Format: The API responds with JSON format.

Status Codes:

- 200 OK: Password reset successful.
- 400 Bad Request: Missing required username, email, or new password.
- 404 Not Found: User does not exist.
- 405 Method Not Allowed: Only POST method is allowed.
- 500 Internal Server Error: An error occurred during the password reset process.

API Name: User Balance Update

Description: The User Balance Update API enables the adjustment of user balances within the web application. It allows for the incremental updating of user balances based on their user ID. Upon receiving the user ID and validating an active session, the API increments the user's balance by a predefined amount in the database. It then retrieves the new balance and updates the session data accordingly.

Endpoints:

URL: /update_balance.php

HTTP Method: POST

Request Parameters:

- user_id (integer): The unique identifier of the user whose balance is being updated.

Response Format: The API responds with JSON format.

Status Codes:

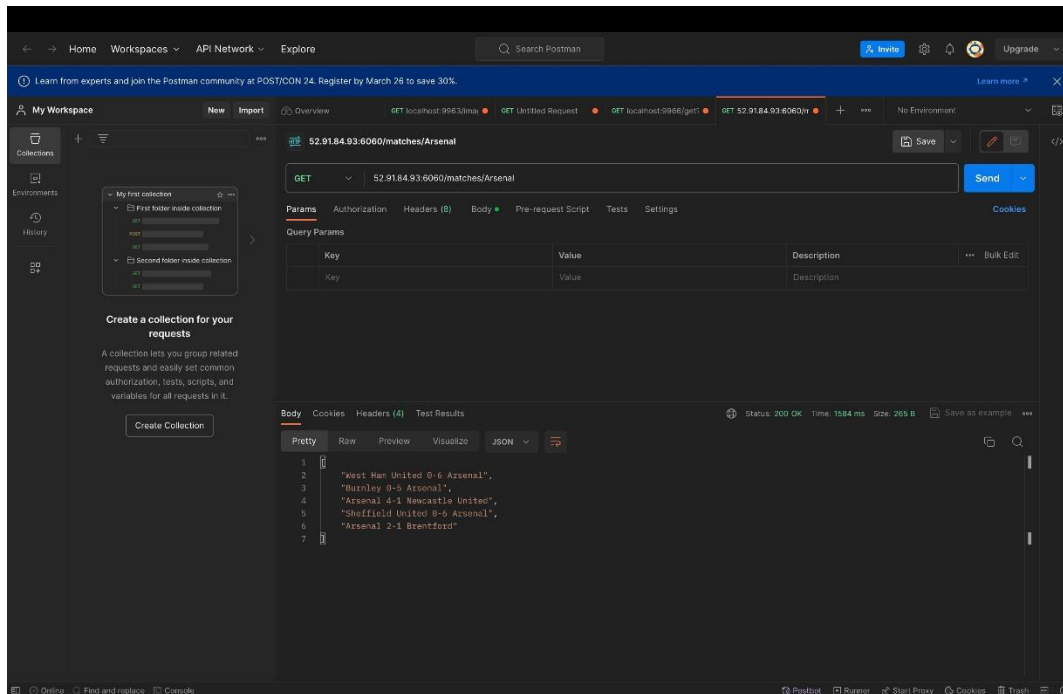
- 200 OK: The balance was successfully updated.

Assignment 2 COS30049 – Group 1.3

- 400 Bad Request: Missing required user ID or session.
- 405 Method Not Allowed: Only POST method is allowed.

API Name: Team's performance API

The URL is varied because of the public IP of the AWS server, deployed in port 6060. The typical URL is: `http://52.91.84.93:6060/matches/[team_name]`



Method: GET

Request parameters: The team name is the only parameter required by the API, it is represented at the end of the GET path, like `http://52.91.84.93:6060/matches/Arsenal`. For teams with space between words, the name is formatted with the '-' replaced for space, like `http://52.91.84.93:6060/matches/Manchester-City`.

Response: If the response status is 200, the content is an array containing the most recent 5 matches of the team requested. A sample of an API call is demonstrated below.

API Name: Blockchain

Method: GET

Request parameters:

- `card_id` (str): A string contains the id of the card. Because the card ID is unique for each card, the token ID, therefore, will certainly be unique.
- `user_id` (int): The user id of the user who buys the card

Response: If the response status is 200, the blockchain system will mint the card. Another support function is `get_token_id_from_card_id` will be used to get the token ID based on the card ID and will return the token ID of the card and save it to the database. This token ID is used to verify the transactions, while the mint function is used to ensure there is not any card bought more than one time.

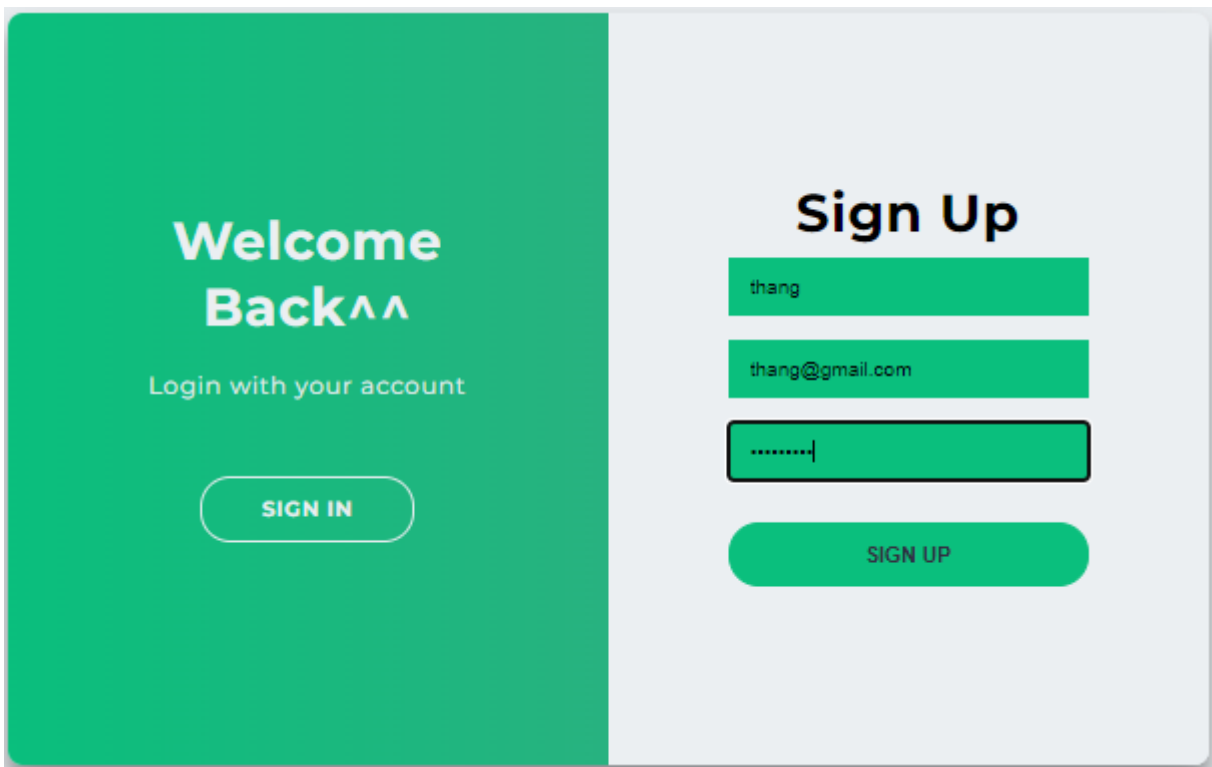
5. Function Description

Functionality Name: User Authentication and Registration

Purpose: The User Authentication and Registration functionality enables users to securely log in and register for access to the Digital Asset Listings and Account Management feature on the blockchain-based platform. With user authentication, existing users can securely log in to their accounts using their username and password, while registration allows new users to create an account and gain access to the platform's features. This functionality ensures that only authenticated users can view and interact with the list of digital assets available for trading.

Use case 1: User Sign Up

Step 1: Type your Username, Email and Password to register



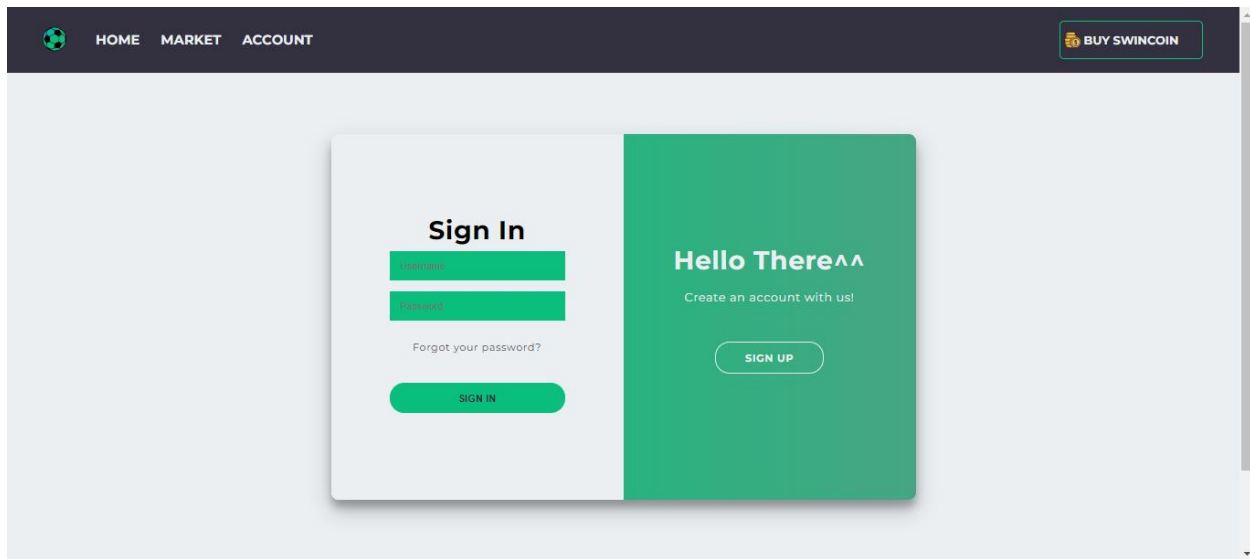
The image shows a user interface for a blockchain-based platform. It is divided into two main sections. The left section has a green background and contains the text "Welcome Back^^" in white, followed by "Login with your account" in a smaller white font, and a white "SIGN IN" button. The right section has a light gray background and contains the title "Sign Up" in bold black font. Below the title are three input fields: the first contains the text "thang", the second contains "thang@gmail.com", and the third contains a masked password ".....". Below these fields is a green "SIGN UP" button.

Step 2: If you successfully sign up, there will be a successful message

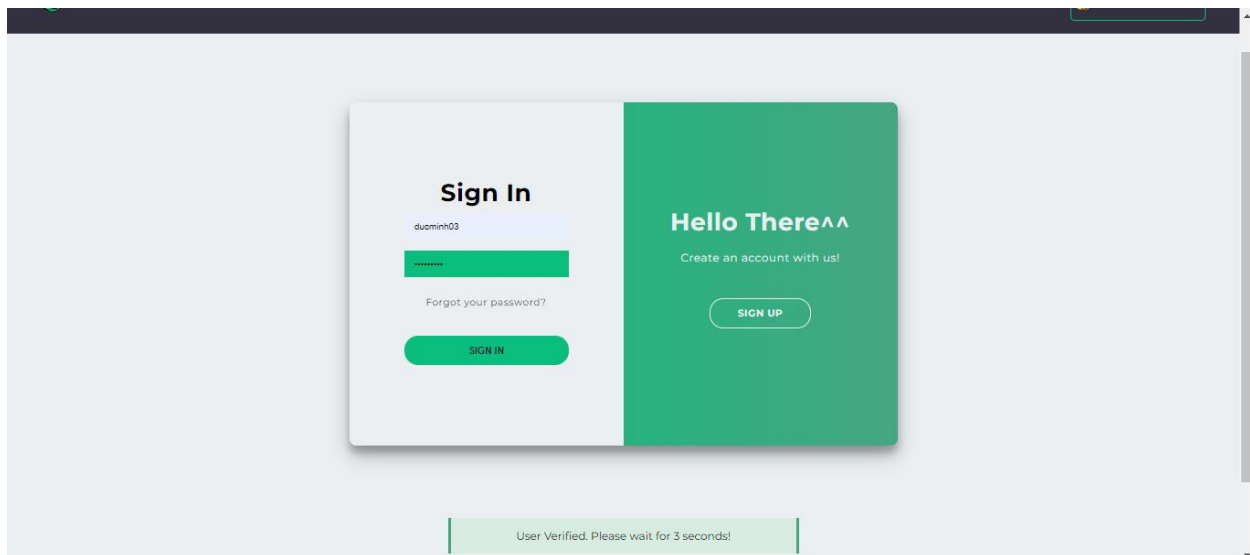
User successfully registered. Please log in with your new account!

Use case 2: User Login

Step 1: Navigate to Market or Account page

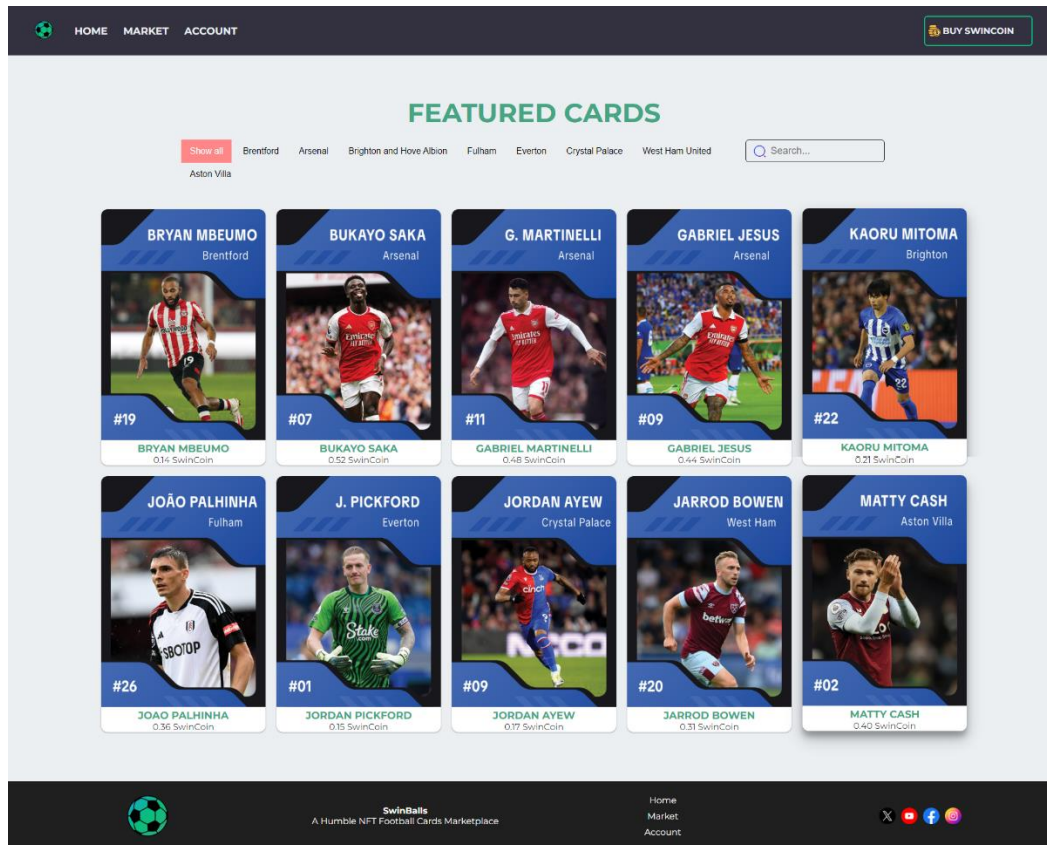


Step 2: Type Username and Password then click 'Sign In' button



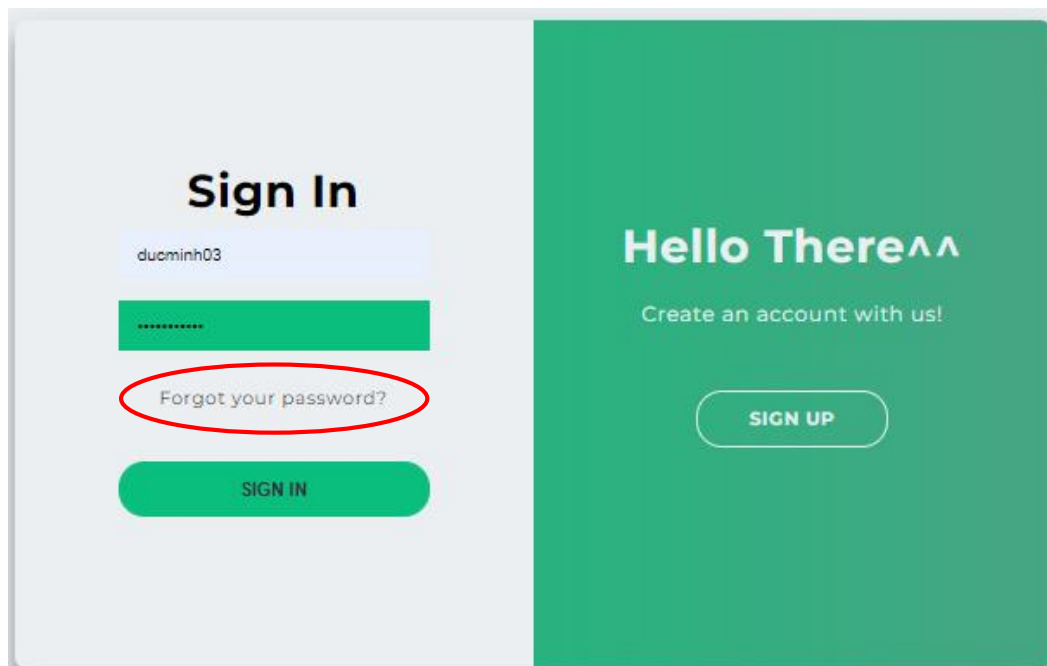
Step 3: If you enter the correct Username and Password that you already registered, there will be a confirmation message and you will have access to the Market and Account page

Assignment 2 COS30049 – Group 1.3

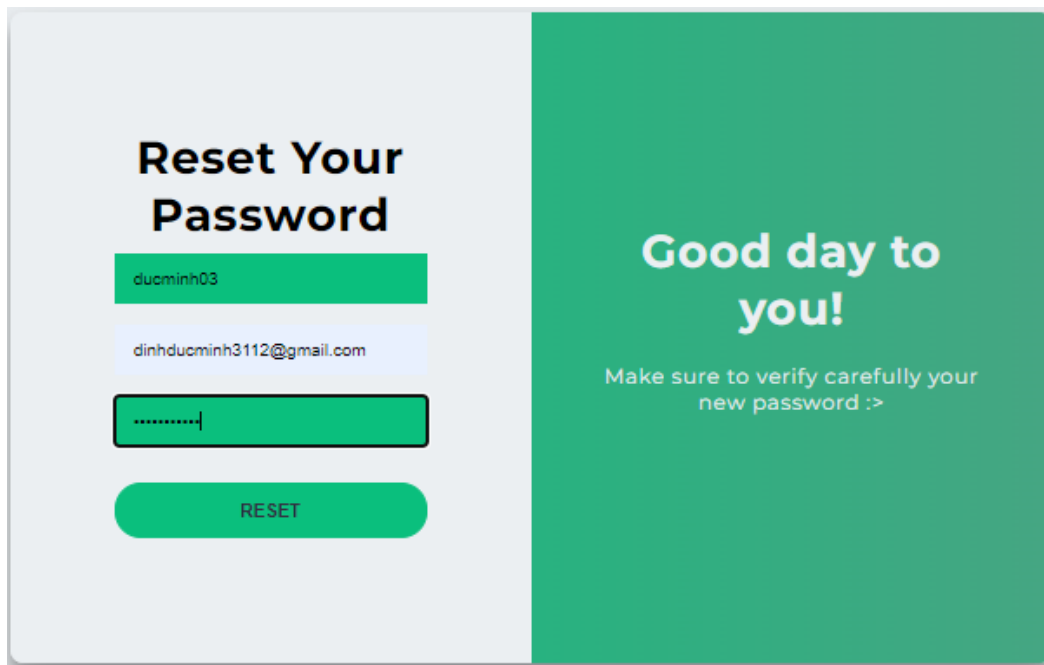


Use case 3: User Forgot Password

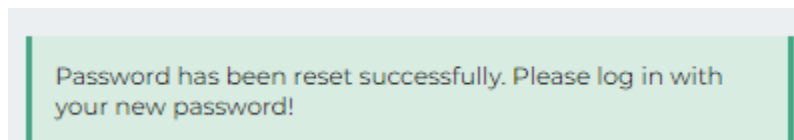
Step 1: Click on 'Forgot your password?'



Step 2: Type your Username, Email and New Password then click 'Reset'

A password reset form with a light gray background on the left and a green background on the right. The left side has the title "Reset Your Password" in bold black text. Below it are three input fields: a green field with "ducminh03", a light blue field with "dinhducminh3112@gmail.com", and a green field with ".....". Below these is a green "RESET" button. The right side has the text "Good day to you!" in white, followed by "Make sure to verify carefully your new password :->" in white.

Step 3: If your Password reset successfully, the website will prompt a successful message

A green rectangular message box with a thin green border. It contains the text "Password has been reset successfully. Please log in with your new password!" in a dark green font.

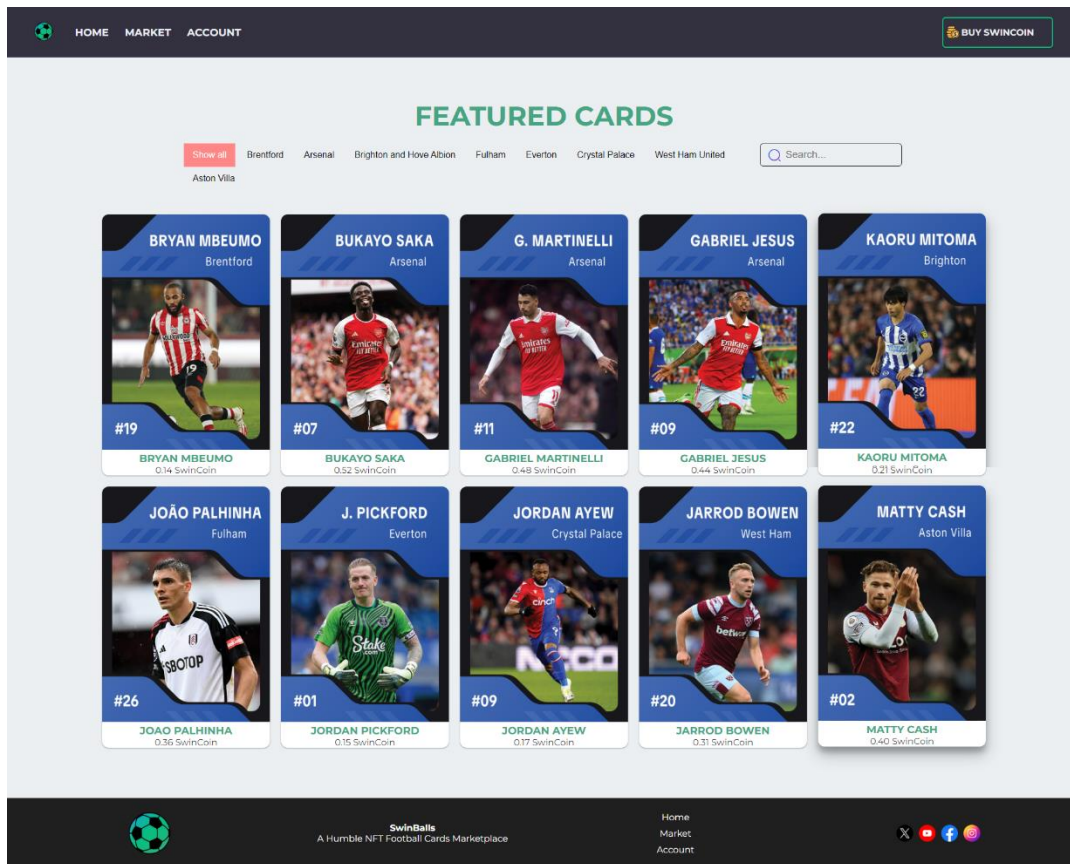
Functionality Name: Digital Asset Listings

Purpose: With the help of this feature, users can see an extensive list of digital assets that are available for trading on the blockchain-based platform. The purpose of Digital Asset Listings delineates the most important data related to each asset, including the image, club, player name, and NFT token ID. By offering this function, users can quickly examine the assets that are available and decide on their trading activities. The card information will be stored in the database including NFT Token ID, image, player name, club, position, age, country and price.

Use case 1: View the NFT list

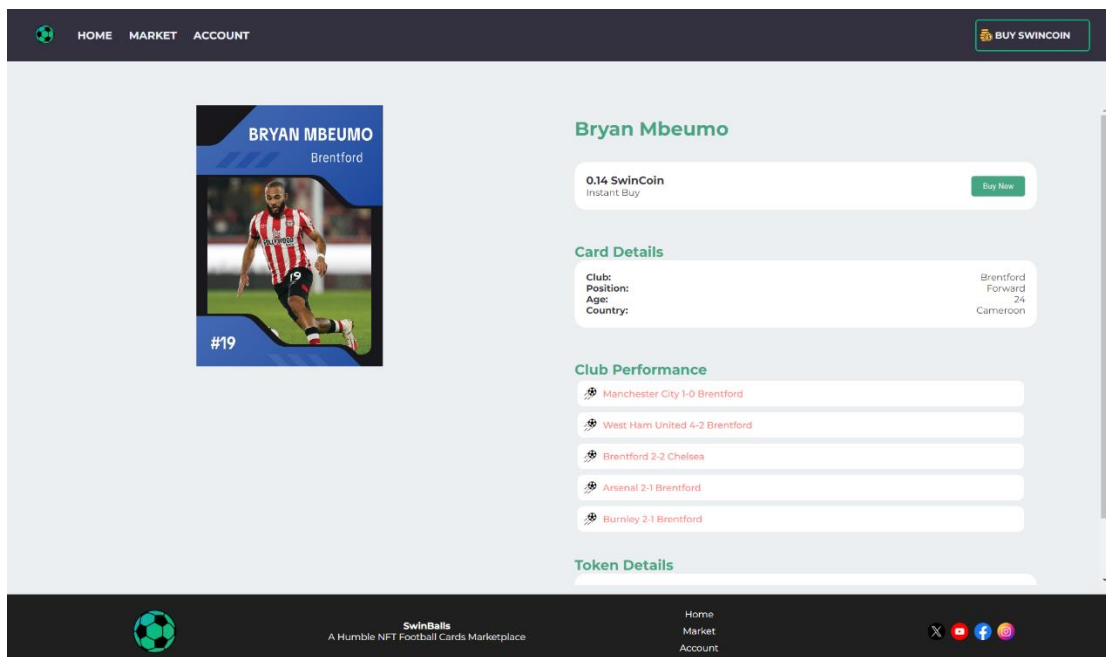
After you successfully log in, you can view the list of NFTs on 'Market' page.

Assignment 2 COS30049 – Group 1.3



Use case 2: View NFT details

To view the NFT details, click on an NFT on 'Market' page



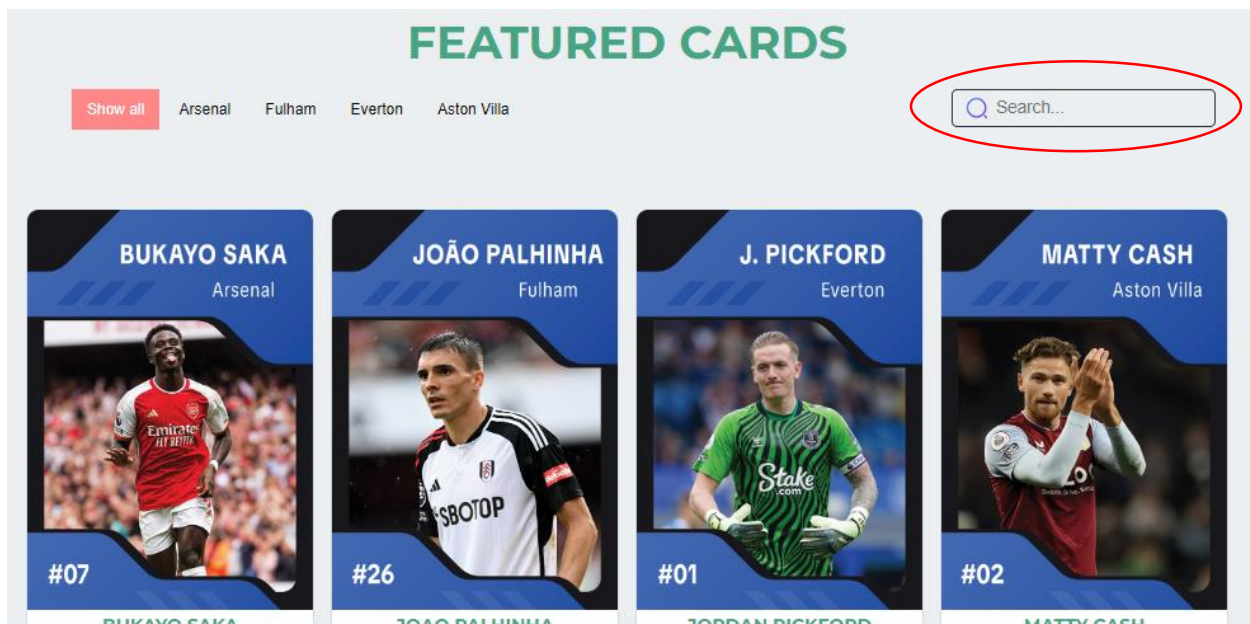
Functionality Name: Search and Filter

Purpose: By adding a search bar and filters, this feature improves user experience by making it easier for users to find particular players, clubs. Users will get the option to search players by name by using filters to sort assets according to player club. The purpose of this is to give users the opportunity to find relevant assets more quickly and efficiently.

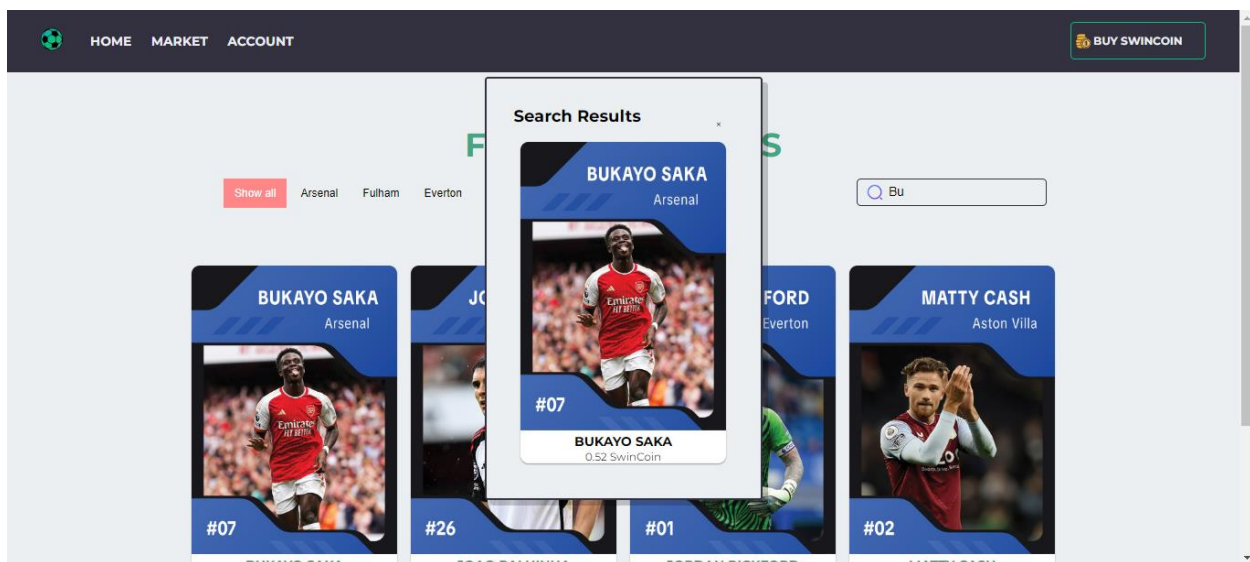
Use case 1: Search for a specific NFT by name

Step 1: Navigate to 'Market' page

Step 2: Click on 'Search' button

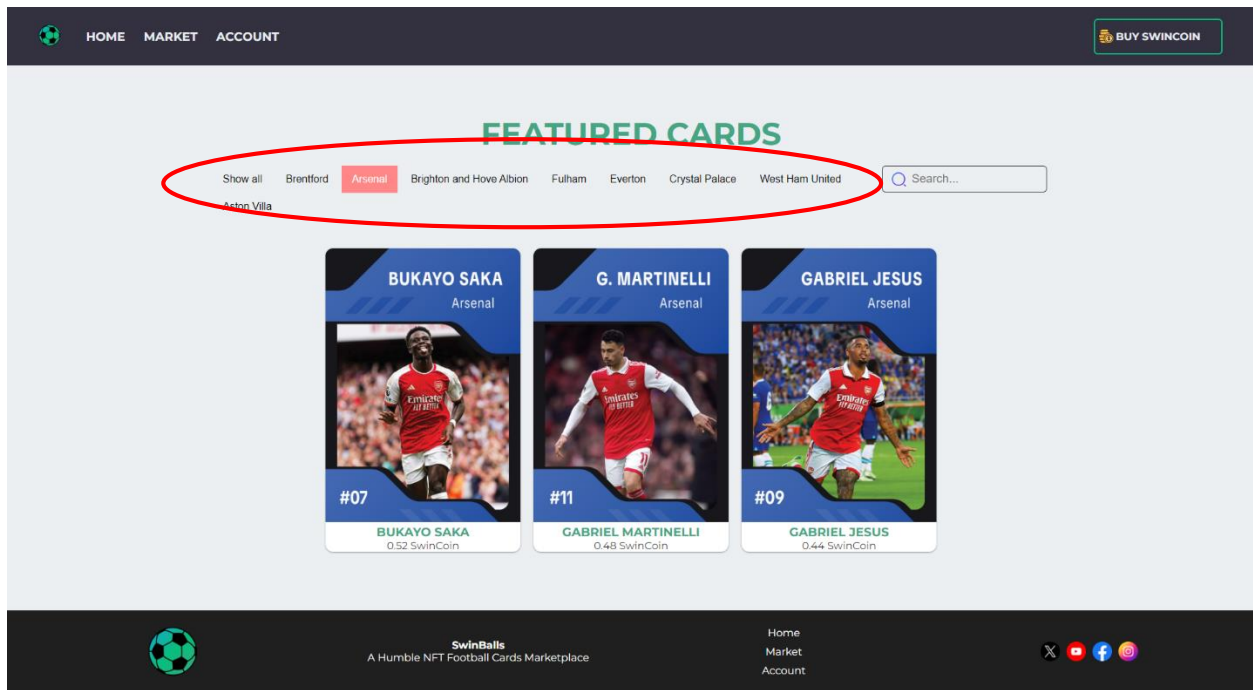


Step 3: Type the player name

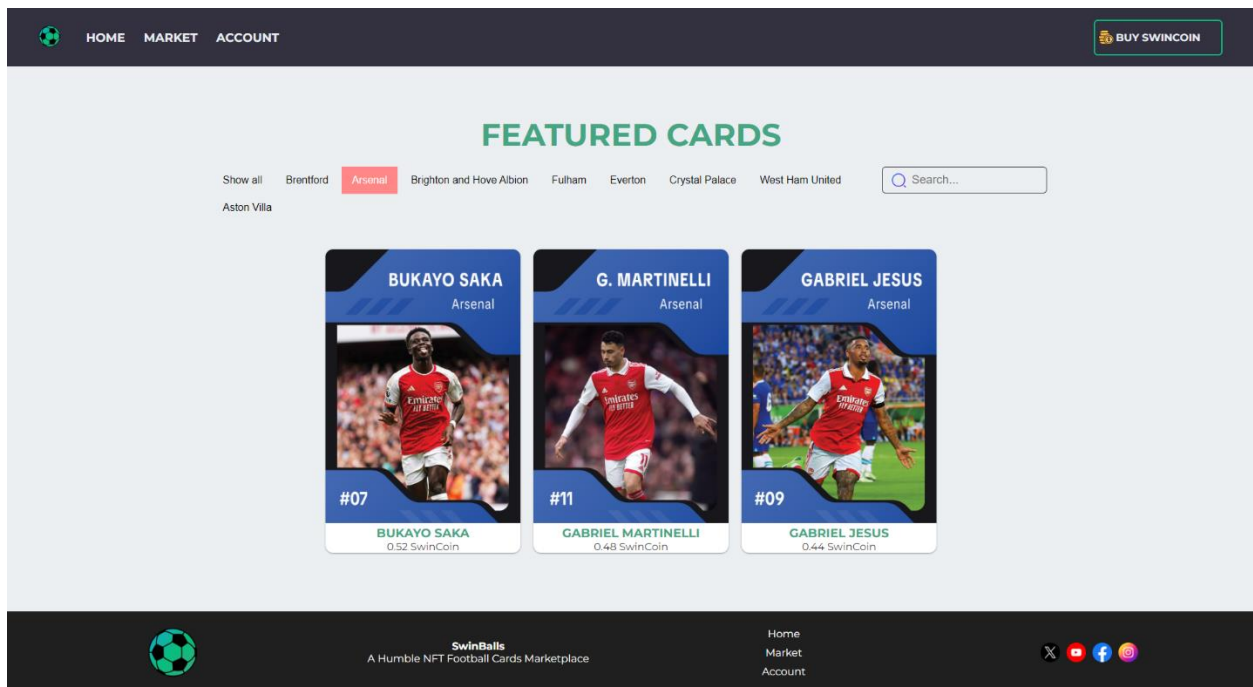


Use case 2: Search NFT by club filter

If you want to search for players that play for a specific club, you click on the ‘Filter’ function



After you click on a specific club, only the players from that club display



Functionality name: Purchasing NFTS

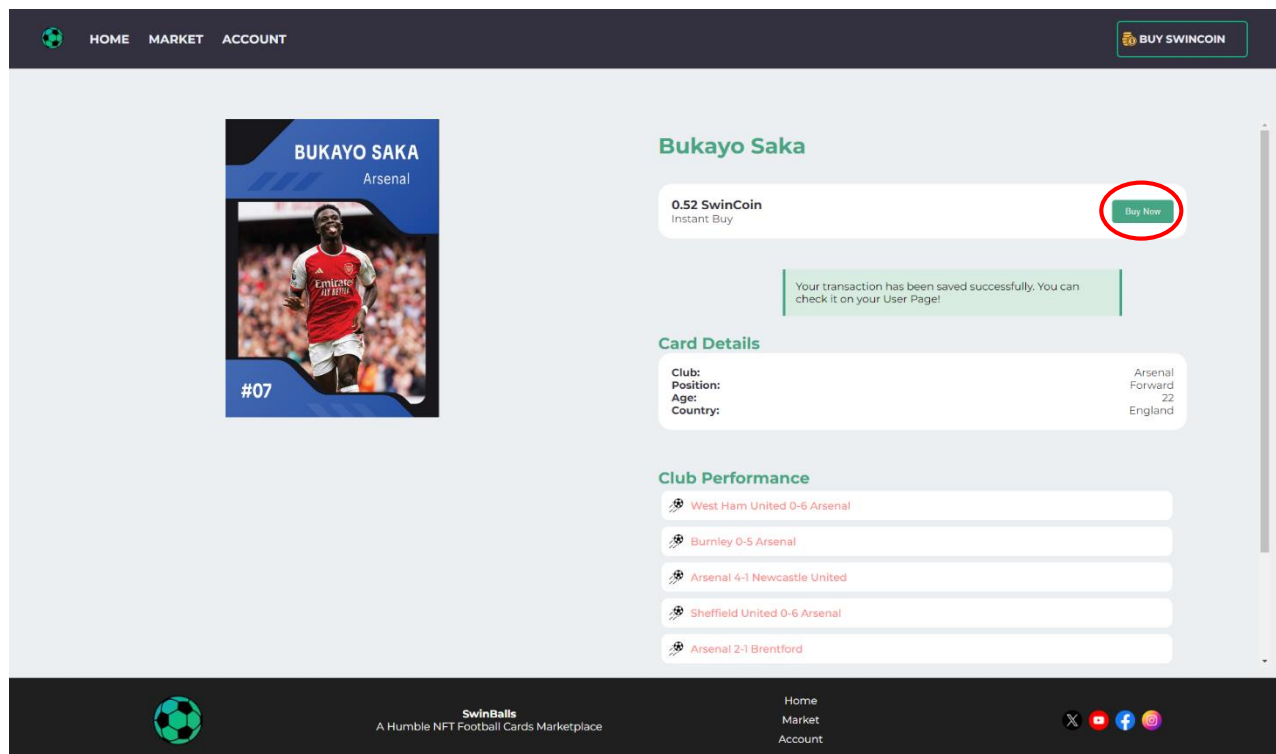
Purpose: The Purchasing NFT function is designed to facilitate peer-to-peer trading of NFTs, which stand in for a variety of football player cards. With this trading feature, users can set up a trade by choosing an asset they own and sending a trade proposal to another user in the blockchain network. Then other users can see and trade their cards. The trade procedure is regulated by the use of smart contracts to ensure the transactions go well and guarantee the safe transfer of assets between the parties to the transaction. If the user does not have enough Swincoin for the card they are seeing and wanting to buy, the 'Buy Now' will not appear and there will be an error message in its place. This feature adds to the dynamic ecosystem of digital asset trading by promoting user interaction and exchange through the facilitation of NFT trading.

Use case 1: User want to buy an NFT

Step 1: Navigate to 'Market' page

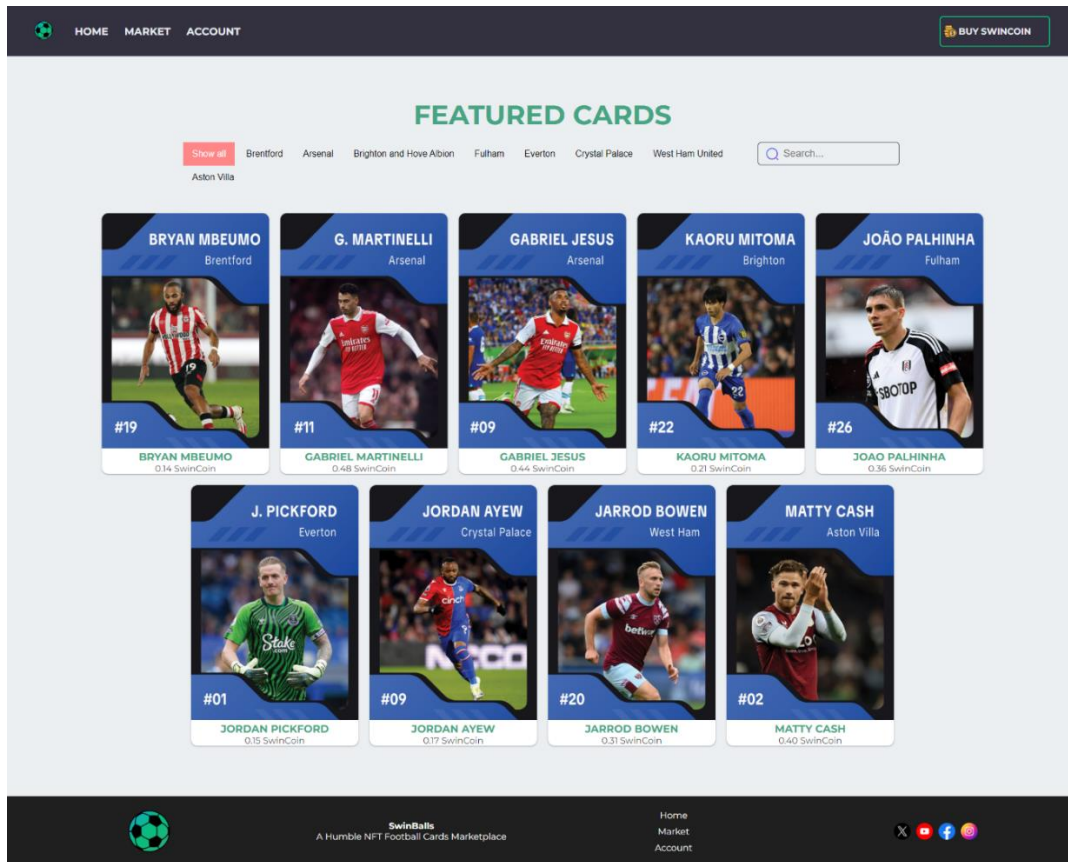
Step 2: Choose an NFT you want to buy

Step 3: Click 'Buy Now'. If the NFT have been bought, there will be a confirmation message below



The NFT that you have bought have been removed from the Market

Assignment 2 COS30049 – Group 1.3

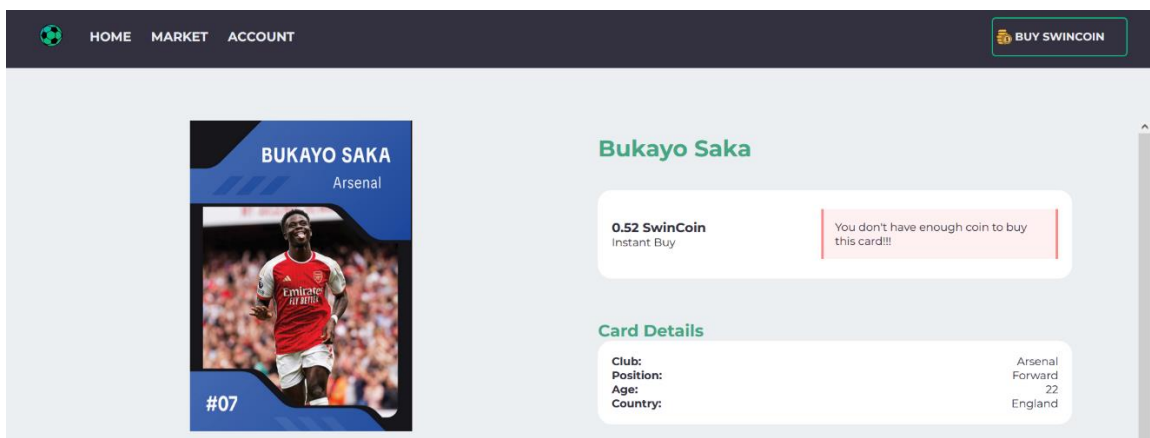


Use case 2: User do not have enough Swincoin to buy an NFT

Step 1: Navigate to 'Market' page

Step 2: Choose an NFT you want to buy

Step 3: The button is replaced with a message that will prompt the user to check if they do not have enough balance. And the user will have to navigate back to the Account page to buy more Swincoin and then proceed to go back to the NFT they want.



Functionality name: View Transaction History

Purpose: The purpose of this function is to give users access to the transactional activity linked to every NFT traded on the blockchain platform. This feature provides a thorough history of previous transactions, complete with important details like the NFT ID, trade status, price, buyer and seller names, and transaction date. Users can track the lifecycle of NFTs, keep an eye on trade activity, and gain insightful knowledge about market dynamics and trends by gaining access to this transaction history.

Use case: User want to view their transaction history

Step 1: Navigate to 'Account' page

Step 2: Click on 'Transaction' below the profile image

The screenshot displays the 'ACCOUNT' page of the SwinBalls marketplace. At the top, there are navigation links for 'HOME', 'MARKET', and 'ACCOUNT', along with a 'BUY SWINCOIN' button. Below the navigation bar, a row of five football player NFT cards is shown, including Lionel Messi, Cristiano Ronaldo, James Rodriguez, Gerard Pique, and Robin van Persie. Underneath the cards is the user's profile section for 'A Random User' (ID: 123456), which includes a profile picture and a menu with 'Personal Details', 'Transactions' (highlighted with a red circle), 'Log Out', and 'Reset Password'. Below the profile section, the 'TRANSACTION HISTORY' is displayed as a table with one transaction entry. The table has columns for Token ID, Card ID, User ID, Purchase Date, Price, and Card Thumbnail. The transaction entry shows Token ID 3, Card ID TWf0dHkgQ2FzaCATIEFzdG9uIFZpbGxhIEZD, User ID 15, Purchase Date 2024-03-27, Price 0.40 SwinCoin, and a thumbnail of a Matty Cash Aston Villa football card (#02). At the bottom of the page, there is a footer with the SwinBalls logo, the text 'A Humble NFT Football Cards Marketplace', and navigation links for 'Home', 'Market', and 'Account', along with social media icons for Twitter, YouTube, Facebook, and Instagram.

Token ID	Card ID	User ID	Purchase Date	Price	Card Thumbnail
3	TWf0dHkgQ2FzaCATIEFzdG9uIFZpbGxhIEZD	15	2024-03-27	0.40 SwinCoin	

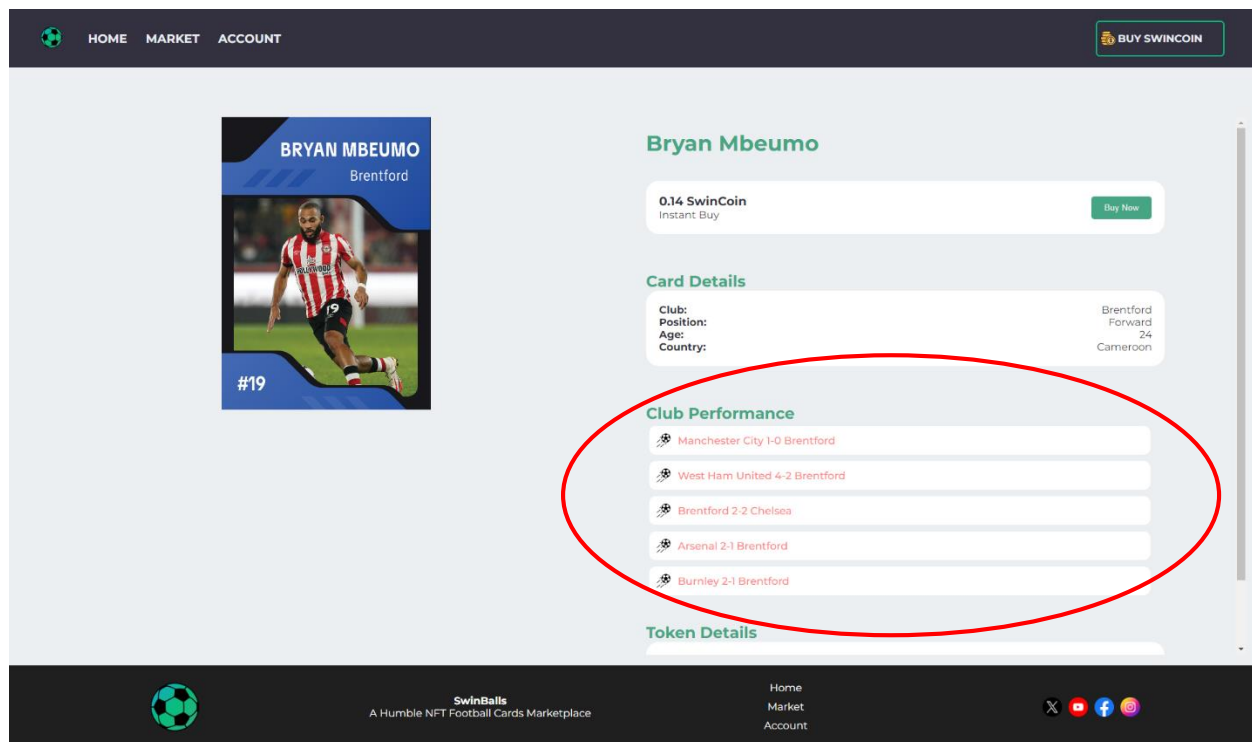
Functionality name: Track Club Performance in real-time (Cutting-edge)

Purpose: The purpose of this functionality is to provide users with up-to-the-minute insights into the performance of the clubs, leveraging real-time data to dynamically adjust the prices of associated NFTs. By integrating cutting-edge technologies and real-time analytics, this functionality enables users to track club performance. These insights are then utilized to dynamically adjust the prices of NFTs associated with the clubs, ensuring that the NFT market reflects the current value and sentiment of the sports clubs. This functionality enhances user engagement, promotes transparency in NFT pricing, and enables more informed decision-making for investors and collectors in the “Swinball” NFT market.

Use case: Users want to see the club's performance to decide on purchases

Step 1: Navigate to ‘Market’ page

Step 2: Choose an NFT you want to track their club's performance



Functionality name: Create an NFT

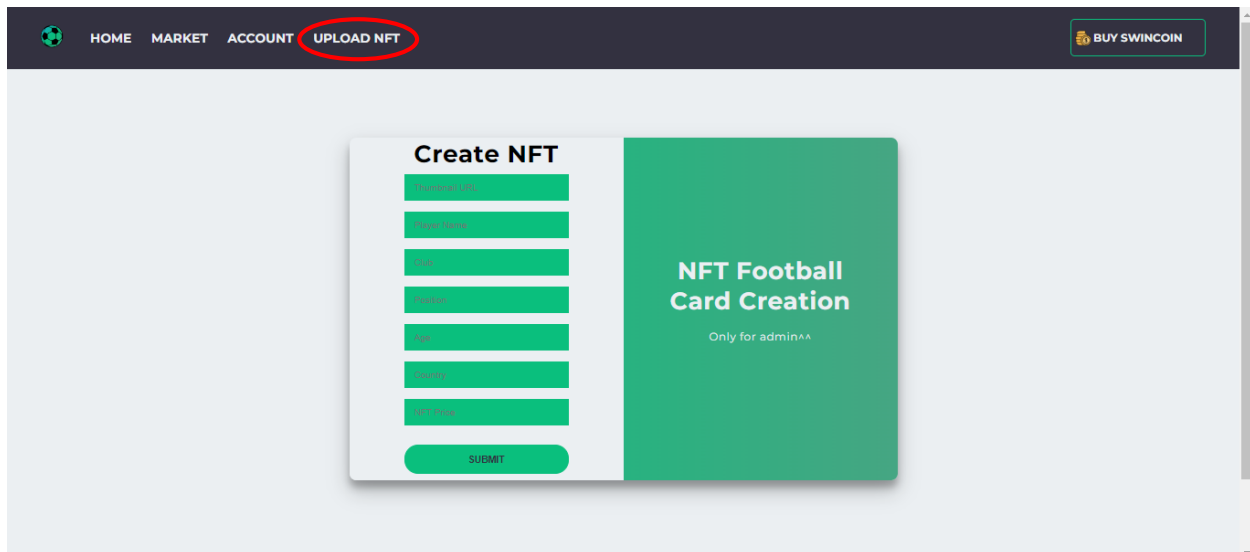
Purpose: The purpose of the "Create an NFT" functionality is to empower authorized administrators with the capability to generate new NFTs within the platform. By limiting access to only administrators, the functionality maintains control over the creation of unique digital assets, safeguarding against unauthorized or inappropriate content being introduced into the platform's ecosystem.

Use case: Admin want to create an NFT

Assignment 2 COS30049 – Group 1.3

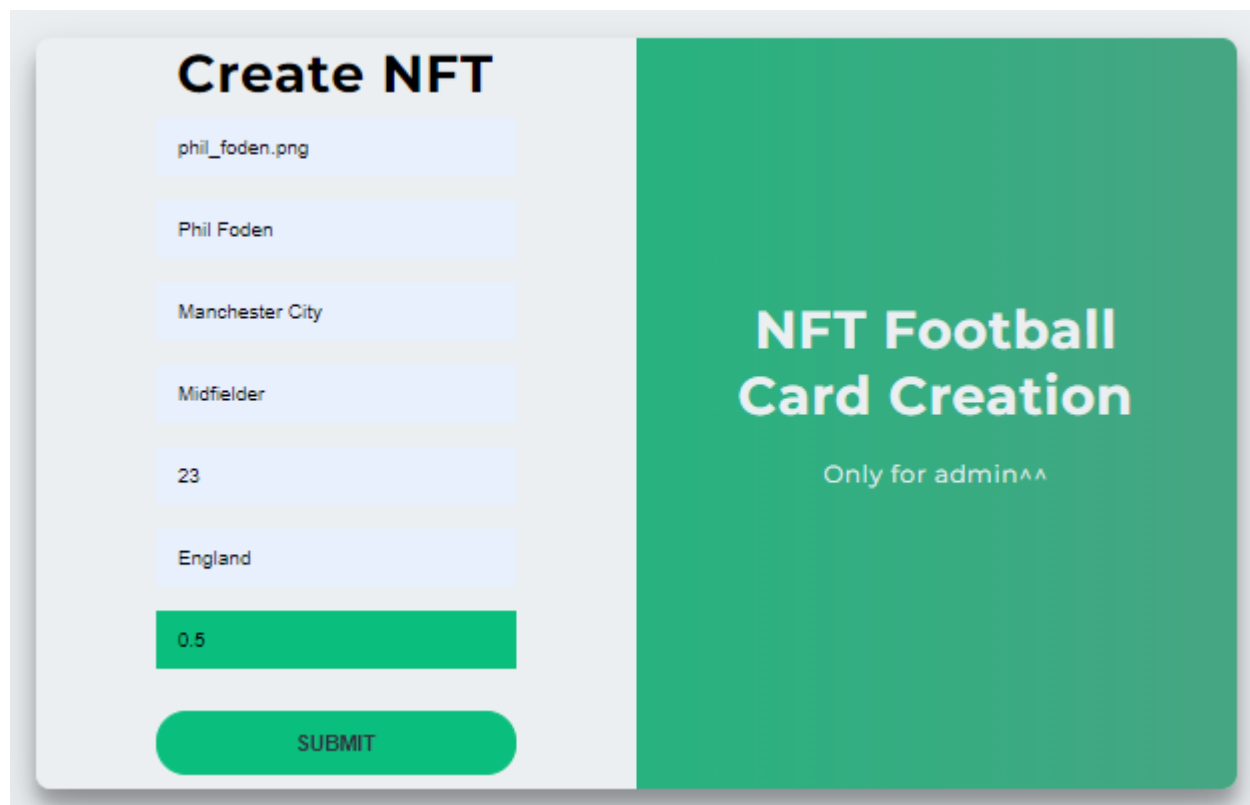
Step 1: You must log in as an admin account

Step 2: Click on 'Upload NFT' page



The screenshot shows a web application interface. At the top, there is a dark navigation bar with links: HOME, MARKET, ACCOUNT, and UPLOAD NFT (which is circled in red). A 'BUY SWINCOIN' button is visible in the top right corner. The main content area features a 'Create NFT' form on the left and a large green panel on the right. The form has the following fields: 'Upload File' (with a file icon), 'Player Name', 'Club', 'Position', 'Age', 'Country', and 'NFT Price'. A 'SUBMIT' button is at the bottom of the form. The green panel on the right contains the text 'NFT Football Card Creation' and 'Only for admin^^'.

Step 3: Fill in the NFT information



The screenshot shows the 'Create NFT' form with the following information filled in: 'phil_foden.png' in the 'Upload File' field, 'Phil Foden' in the 'Player Name' field, 'Manchester City' in the 'Club' field, 'Midfielder' in the 'Position' field, '23' in the 'Age' field, 'England' in the 'Country' field, and '0.5' in the 'NFT Price' field. The 'SUBMIT' button is at the bottom of the form. The green panel on the right remains the same, displaying 'NFT Football Card Creation' and 'Only for admin^^'.

Step 4: Then, click 'Submit'

Create NFT

phil_foden.png

Phil Foden

Manchester City

Midfielder

23

England

0.5

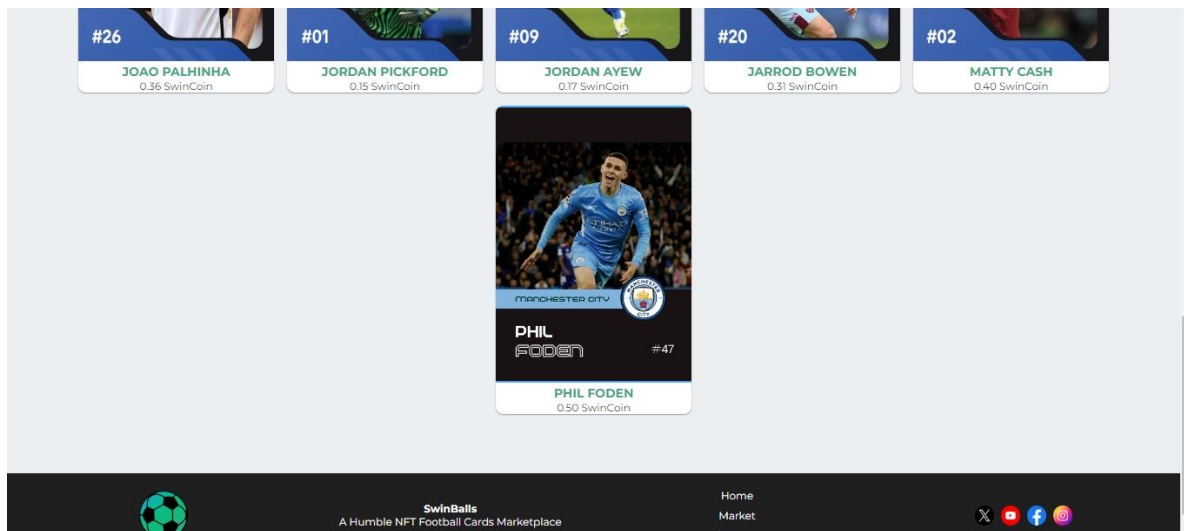
SUBMIT

NFT Football Card Creation

Only for admin^^

The latest NFT has been created in the database with the ID: UGhpbCBGb2Rlbk1hbmNoZXNOZXlqQ2l0eQ==

A confirmation message will prompt with the ID generated by the combination of player name and club then converted to base64_encode.



New NFT have been uploaded to the Market

6. Project Deployment Instruction

1. Tools and version:

- PHP: 8.2.12
- XAMPP: v3.3.0
- Ipython: 8.12.0
- Ipython_genutils: 0.2.0
- Msgpack-python: 1.0.3
- Python: 3.11.4
- Uvicorn: 0.23.2
- Fastapi: 0.103.1
- BeautifulSoup4: 4.12.2
- Truffle: v5.11.5
- Ganache: v7.9.1
- Solidity: v0.5.16 (solc-js)
- Node: v20.11.1
- Web3.js: v1.10.0

2. Installation Process:

- Download and unzip the file class01-group-1.3.zip.
- Move the folder 'Source Code' into ../xampp/htdocs to run it locally with XAMPP
- Install the tools on the list above for the program to run properly
- Open XAMPP application and start the Apache module and open up your browser and look for localhost/Source Code/index.php to see the homepage of Swinball and have a first look at the interface of the website.

3. Database Configuration:

We use MySQL database, so this is our query to create our three tables (Or you can use queries in Database_Query_30049.txt and Demo_data.txt):

```
CREATE TABLE users (  
    user_id int(20) NOT NULL AUTO_INCREMENT,  
    username varchar(50) NOT NULL,  
    password varchar(50) NOT NULL,  
    email varchar(100) NOT NULL,  
    date_joined DATE NOT NULL,  
    balance DECIMAL(10,2),  
    PRIMARY KEY (user_id),  
    UNIQUE KEY email (email)
```

);

```
CREATE TABLE cards (  
    card_id VARCHAR(255) PRIMARY KEY,  
    image_url VARCHAR(255),  
    player_name VARCHAR(50),  
    club VARCHAR(50),  
    position VARCHAR(20),  
    age INT(10),  
    country VARCHAR(20),  
    price DECIMAL(10,2)  
);
```

```
CREATE TABLE transactions (  
    token_id INT(50) PRIMARY KEY,  
    card_id VARCHAR(255),  
    user_id INT(20),  
    purchase_date DATE,  
    price DECIMAL(10,2),  
    FOREIGN KEY (card_id) REFERENCES cards(card_id),  
    FOREIGN KEY (user_id) REFERENCES users(user_id)  
);
```

```
CREATE INDEX idx_card_id ON cards (card_id);
```

```
CREATE INDEX idx_transaction_id ON transactions (token_id);
```

And below is the query to insert the demo dataset used for the demo website:

```
INSERT INTO cards (card_id, image_url, player_name, club, position, age, country, price)  
VALUES
```

('QnVrYXlvIFNha2EgLSBBcnNlbnFs', 'bukayo_saka.jpg', 'Bukayo Saka', 'Arsenal', 'Forward', 22, 'England', 0.52),

('Sm9hbyBQYWxoaW5oYSAtdEZlbnGhhbSBGQw==', 'joao_palhinha.jpg', 'Joao Palhinha', 'Fulham', 'Midfielder', 28, 'Portugal', 0.36),

('Sm9yZGFuIFBpY2tmb3JkIC0gRXZlcnRvbiBGQw==', 'jordan_pickford.jpg', 'Jordan Pickford', 'Everton', 'Goalkeeper', 30, 'England', 0.15),

('TWF0dHkgQ2FzaCAtdEFzdG9uIFZpbGxhIEZD', 'matty_cash.jpg', 'Matty Cash', 'Aston Villa', 'Defender', 26, 'Poland', 0.40),

('S2FvcnVNaXRvbWFCcmllnaHRvbmFuZEhvdnVBbGJpb24=', 'kaoru_mitoma.jpg', 'Kaoru Mitoma', 'Brighton and Hove Albion', 'Forward', 26, 'Japan', 0.21),

('R2FicmlleE1hcnRpbmVsbGlBcnNlbnFs', 'gabriel_martinelli.jpg', 'Gabriel Martinelli', 'Arsenal', 'Forward', 22, 'Brazil', 0.48),

('R2FicmlleEplc3VzQXJzZW5hbA==', 'gabriel_jesus.jpg', 'Gabriel Jesus', 'Arsenal', 'Forward', 26, 'Brazil', 0.44),

('SmFycm9kQm93ZW5XZXN0SGFtVW5pdGVk', 'jarrod_bowen.jpg', 'Jarrod Bowen', 'West Ham United', 'Forward', 27, 'England', 0.31),

('QnJ5YW5NYmVlbW9CcmVudGZvcnQ=', 'bryan_mbeumo.jpg', 'Bryan Mbeumo', 'Brentford', 'Forward', 24, 'Cameroon', 0.14),

('Sm9yZGFuQXlld0NyeXN0YWxQYWxhY2U=', 'jordan_ayew.jpg', 'Jordan Ayew', 'Crystal Palace', 'Forward', 32, 'Ghana', 0.17);

4. Blockchain Integration

Step 1: AWS config

Sign in to AWS, create an EC2 Instance:

- Choose Ubuntu AMI.
- Select instance type as t2.micro to stay within the free tier.
- Create a Key Pair and save it in a folder.

Edit Security Groups, Inbound rules:

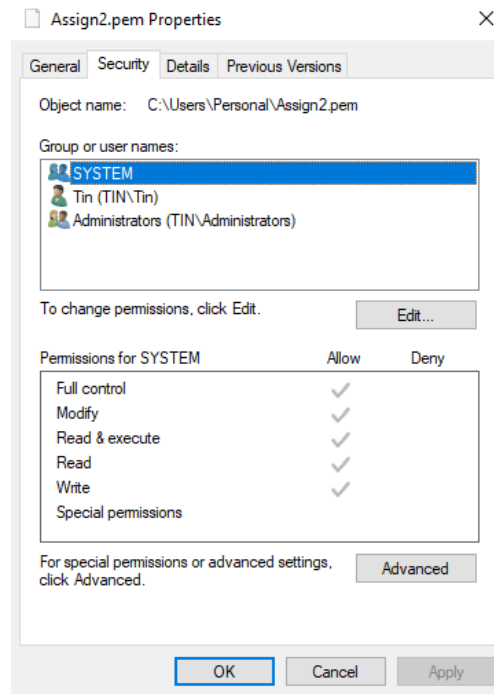
- Custom TCP(22) from Anywhere
- Custom TCP(80) from Anywhere
- Custom TCP(8545) from Anywhere
- Custom TCP(30303) from Anywhere
- Custom TCP(443) from Anywhere

Assignment 2 COS30049 – Group 1.3

Launch Instance.

Click on Connect Instance. Go to SSH, and copy the SSH link.

Go to the folder where the Key Pair was saved. Right-click on the key properties and set permissions as follows:



On the address bar of the folder, press cmd to open Terminal.

Enter the SSH link copied from AWS and run it. If Ubuntu appears, it's okay.

Step 2: Ethereum installation

Enter `sudo su` to switch to root privilege.

Enter: `cd ../../` to navigate outside the main folder.

Enter `ls` to view all files and folders. If you see the `opt` file, then it's correct.

Enter `cd opt/`

Enter `mkdir ethPoA` to create the `ethPoA` folder.

Enter `cd ethPoA` to enter the folder.

Enter `mkdir node0` to create the `node0` folder.

Enter `cd node0` to enter the folder.

Enter `sudo add-apt-repository -y ppa:ethereum/ethereum`

Assignment 2 COS30049 – Group 1.3

Enter `sudo apt-get update`

Enter `sudo apt-get install ethereum`

Enter `geth --datadir "./data" account new`

After this step, it will prompt to create a password, for example, 123456. Then, you will see the Wallet address. Note that you should save the address somewhere (e.g., Notepad) to remember it later.

Enter `ls` to display the folder, you will see the data folder.

Now, enter: `vim password.txt`

Press `i` to insert.

Enter the password you just created here.

Then press the Esc key, then enter `:wq` to save and exit.

Similarly, `vim ethpoa.json`

Replace with your Wallet address with the underline:

```
{
  "config": {
    "chainId": 9999,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip150Hash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "clique": {
      "period": 5,
      "epoch": 30000
    }
  }
}
```

[illegible]

Press Esc and :wq to save and exit.

After successful execution, enter: (Note: also need to edit Wallet address and Private IP address from the AWS Instance of the group)

Assignment 2 COS30049 – Group 1.3

```
geth --networkid 9999 --datadir ./data --port 30303 --ipcdisable --syncmode full --http --allow-insecure-unlock --http.corsdomain "*" --http.port 8545 --http.addr "172.31.46.93" --unlock 0xA3F3F4Dda6beA8C9bAd0942b8bD5825366A89563 --password ./password.txt --mine --http.api personal,admin,db,eth,net,web3,miner,shh,txpool,debug,clique --ws --ws.addr 0.0.0.0 --ws.port 8546 --ws.origins '*' --ws.api personal,admin,db,eth,net,web3,miner,shh,txpool,debug,clique --maxpeers 25 --miner.etherbase 0xA3F3F4Dda6beA8C9bAd0942b8bD5825366A89563 --miner.gasprice 0 --miner.gaslimit 99999999
```

(Note: This command will run continuously to maintain the server, do not shut down if still in use.)

Step 3: Migrate Blockchain to Server

In your contract folder, run:

```
http://(public I:P):8545
truffle console --network AWS
truffle compile
truffle migrate
```

Step 4: Team's performance API

Do similar step 1 to create an instance on AWS, then ssh to it. In the second server:

- Install miniconda:

```
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm -rf ~/miniconda3/miniconda.sh
```

- Logout and ssh again to restart

- Install libraries:

```
pip install uvicorn
pip install fastapi
pip install bs4
```

Assignment 2 COS30049 – Group 1.3

- Create python file: `vim topmatch.py`, type i, and paste this before use ESC and `:wq` to exit:

```
from fastapi import FastAPI
import requests
from bs4 import BeautifulSoup
app = FastAPI()
def match(hai):
    score = hai.find('span', class_='match-fixture__score score').text.replace(' ', '')
    home = hai.find_all('span', class_='match-fixture__team-name')[0].abbr['title']
    away = hai.find_all('span', class_='match-fixture__team-name')[1].abbr['title']
    return f"{home} {score} {away}"
def recent_matches(name):
    name = name.replace('-', ' ')
    url = "https://www.premierleague.com/tables"
    response = requests.get(url)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        teams = soup.find_all('td', class_='league-table__team')
        team_list = []
        for team in teams:
            team_name = team.find('span', class_='long')
            team_list.append(team_name.text.strip())
        if name not in team_list:
            return []
        idx = team_list.index(name)
        content = soup.find_all('td', class_='league-table__form form hideMed')[idx]
        for team in [content]:
            hai = team.find_all('a', class_='tooltipContainer linkable tooltip-link tooltip-right')
            match_list = []
```

```
        for i in range(5):
            match_list.append(match(hai[i]))

        return match_list

print(recent_matches('Manchester-United'))

# API for recent matchs

@app.get("/matches/{team_name}")

def get_recent_matches(team_name: str):

    return recent_matches(team_name)
```

Deploy API by running: `uvicorn topmatch:app --host 0.0.0.0 --port 6060`

Now the deployment is completed, you can enjoy searching and collecting your favourite football players' cards!

VI. Conclusion

In conclusion, the Swinball Football Card NFT Trading Platform represents a world of football card collecting by using the blockchain technology to ensure security, transparency, and authenticity in football card transactions. Our platform provides users with a smooth experience, and a user-friendly interface. Though some functions might not seem necessary due to the limited scale of the demo version, it promises to adapt to larger scale with heavier traffic.

The Swinball Football Card NFT is committed to bring a different breeze to the world of football card collecting enthusiasts and connect them with their favourite players and moments in an unprecedented way. This is also a place where people can entrust their money and make a fortune with rare football pieces.

VII. References

Submitting AJAX Forms with jQuery. (n.d.). DigitalOcean Community Tutorials. <https://www.digitalocean.com/community/tutorials/submitting-ajax-forms-with-jquery>

Gol, S. (n.d.). Understanding How a Blockchain Works With PHP and JSON. Medium. <https://medium.com/@simogol/understanding-how-a-blockchain-works-with-php-and-json-ee9305d7d2c1>

Shaggyrec. (n.d.). Building a Simple Blockchain in PHP. DEV Community. <https://dev.to/shaggyrec/building-a-simple-blockchain-in-php-2b8n>

How to Build a Blockchain App. (n.d.). Dapp University. <https://www.dappuniversity.com/articles/how-to-build-a-blockchain-app>

Samuel, J.O. (n.d.). Building Blockchain Applications With Golang Vs Solidity. Medium. <https://medium.com/@jefferyokesamuel1/building-blockchain-applications-with-golang-vs-solidity-6f3c2c38a5b1>

Envoy_. (n.d.). Build Your First Smart Contract Using Solidity. DEV Community. https://dev.to/envoy_/build-your-first-smart-contract-using-solidity-111p

Solidity101. (n.d.). Smart Contract Development: Learning Solidity and Becoming a Blockchain Developer. Medium. <https://medium.com/@solidity101/smart-contract-development-learning-solidity-and-becoming-a-blockchain-developer-c9b3ac14cb5e>

Student Name	Student ID
Dinh Duc Minh	103487185
Tran Viet Thang	103846676
Doan Minh Hieu	104168106
Le Nho Bach	104221795
Truong Minh Son	104221795

Final Word Count: 4922