****

**MURANG’A UNIVERSITY OF TECHNOLOGY**

**Name**: ODHIAMBO VICTOR AMON

**Reg**: SC212/0601/2022

**Program**: BSC SOFTWARE ENGINEERING

**Year of study**: SECOND YEAR

**Academic Year:** 2023/2024

**Unit**: PROGRAMMING AND DATABASE PRACTICUM CORE

**Unit code**: SCS203

# DECLARATION

I certified that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any university ,and to the best of my knowledge and belief, it does not contain any material previously published or written by another person. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and interlibrary loans and for the title and abstract to be made available to outside organizations.

Signature of candidate…………………. Date signed………………………….

Name of candidate………………………………………………………

Reg no …………………………………………………...

Countersigned by:

Signature of supervisor(s)/ Advisor(s)………………. Date signed………………...

Name of supervisor(s)/ Advisor(s)…………………………………………………..

# TABLE OF CONTENTS

Contents

[DECLARATION 2](#_Toc164254695)

[TABLE OF CONTENTS 3](#_Toc164254696)

[TABLE OF FIGURES 4](#_Toc164254697)

[ACKNOWLEDGEMENT 6](#_Toc164254698)

[INTRODUCTION 6](#_Toc164254699)

[EXPLANATION OF CONTENTS 7](#_Toc164254700)

[ABSTRACT 7](#_Toc164254701)

[CHAPTER 1 7](#_Toc164254702)

[1.1 BACKGROUND 7](#_Toc164254703)

[1.1.1 Existing systems 8](#_Toc164254704)

[1.1.2 Proposed systems 9](#_Toc164254705)

[1.2 OBJECTIVES 9](#_Toc164254706)

[1.3 PURPOSE, SCOPE, APPLICABILITY 9](#_Toc164254707)

[1.3.1 PURPOSE 9](#_Toc164254708)

[1.3.2 SCOPE 9](#_Toc164254709)

[1.3.3 APPLICABILITY 10](#_Toc164254710)

[CHAPTER 2: SURVEY OF TECHNOLOGIES 11](#_Toc164254711)

[Overview 11](#_Toc164254712)

[Rationale 11](#_Toc164254713)

[Roles 11](#_Toc164254714)

[Programming languages and data stores 11](#_Toc164254715)

[Frameworks and Libraries 11](#_Toc164254716)

[Tools 12](#_Toc164254717)

[CHAPTER 3: REQUIREMENTS AND ANALYSIS 13](#_Toc164254718)

[3.1 PROBLEM DEFINITION 13](#_Toc164254719)

[3.2 REQUIREMENT SPECIFICATION 13](#_Toc164254720)

[3.2.1 HARDWARE REQUIREMENTS 14](#_Toc164254721)

[3.2.2 SOFTWARE REQUIREMENTS 14](#_Toc164254722)

[3.2.3 FUNCTIONAL REQUIREMENTS 15](#_Toc164254723)

[3.2.4 NON-FUNCTIONAL REQUIREMENTS 15](#_Toc164254724)

[3.3 PRELIMINARY PRODUCT DESCRIPTION 15](#_Toc164254725)

[3.4 CONCEPTUAL MODELS 16](#_Toc164254726)

[SYSTEM FLOWCHART 16](#_Toc164254727)

[ENTITY RELATIONSHIP DIAGRAMS 17](#_Toc164254728)

[CHAPTER 4: SYSTEM DESIGN 18](#_Toc164254729)

[4.1 BASIC MODULES 18](#_Toc164254730)

[4.2 DATA DESIGN 30](#_Toc164254731)

[4.2.1 SCHEMA DESIGN 30](#_Toc164254732)

[4.2.2 DATA INTEGRITY AND CONSTRAINTS 34](#_Toc164254733)

[4.3 PROCEDURAL DESIGN 34](#_Toc164254734)

[4.3.1 LOGIC DIAGRAMS 34](#_Toc164254735)

[4.3.2 DATA STRUCTURES 34](#_Toc164254736)

[4.3.3 ALGORITHMS DESIGN 34](#_Toc164254737)

[4.3.4 USER INTERFACE DESIGN 35](#_Toc164254738)

[4.3.5 SECURITY ISSUES 35](#_Toc164254739)

[4.3.6 TEST CASE DESIGN 35](#_Toc164254740)

[CHAPTER 5: IMPLEMENTATION AND TESTING 36](#_Toc164254741)

[5.1 IMPLEMENTATION APPROACHES 36](#_Toc164254742)

[5.2 CODING DETAILS AND CODE EFFICIENCY 36](#_Toc164254743)

[5.3 TESTING APPROACHES 37](#_Toc164254744)

[5.3.1 UNIT TESTIING 37](#_Toc164254745)

[5.3.2 INTEGRATION TESTING 37](#_Toc164254746)

[5.4 MODIFICATIONS AND IMPROVEMENTS 38](#_Toc164254747)

[5.5 ENVIRONMENT SETUP 38](#_Toc164254748)

[CHAPTER 6: CONCLUSIONS 40](#_Toc164254749)

[6.1 CONCLUSIONS 40](#_Toc164254750)

[6.2 LIMITATIONS OF THE SYSTEM 40](#_Toc164254751)

[6.3 FUTURE DIRECTIONS 40](#_Toc164254752)

[References: 41](#_Toc164254753)

# TABLE OF FIGURES

[Figure 1 Login 16](#_Toc164252179)

[Figure 2 login code 17](#_Toc164252180)

[Figure 3 invalid login credentials 18](#_Toc164252181)

[Figure 4 signup code 19](#_Toc164252182)

[Figure 5 signup page 20](#_Toc164252183)

[Figure 6 User Dashboard 21](#_Toc164252184)

[Figure 7 Product info upload 22](#_Toc164252185)

[Figure 8 Product list 23](#_Toc164252186)

[Figure 9 Listing a product for auctioning 23](#_Toc164252187)

[Figure 10 Auctioning market place 24](#_Toc164252188)

[Figure 11 Bidding process 25](#_Toc164252189)

[Figure 12 seller upload auctioned product 26](#_Toc164252190)

[Figure 13 Transactions page 26](#_Toc164252191)

[Figure 14 Buyer makes funds transactions 27](#_Toc164252192)

[Figure 15 Account details 27](#_Toc164252193)

[Figure 16 Update password interface 27](#_Toc164252194)

[Figure 17 Update password code 28](#_Toc164252195)

[Figure 18 User table Schema 28](#_Toc164252196)

[Figure 19 Auctions table schema 29](#_Toc164252197)

[Figure 20 Bidding table schema 29](#_Toc164252198)

[Figure 21 Property table schema 30](#_Toc164252199)

[Figure 22 Transactions table schema 31](#_Toc164252200)

[Figure 23 Approvals table schema 31](#_Toc164252201)

[Figure 24 Integrity constraint, unique key and validation for field UserID 32](#_Toc164252202)

[Figure 25 MVC approach 34](#_Toc164252203)

[Figure 26 Unit testing 35](#_Toc164252204)

[Figure 27 Integration testing 36](#_Toc164252205)

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my lecturer Tirus Muya who offered me a golden opportunity to do this wonderful project on car rental system, and who has been always there to give advice when necessary. This also helped me in doing a lot of research that I came to know so many things.

Secondly, I would like to thank my parents, siblings and friends who helped me a lot in coming up with this project.

# INTRODUCTION

The online auctioning system represents a dynamic and innovative platform designed to revolutionize the buying and selling experience in the digital realm. With its robust feature set and intuitive user interface, the system offers a seamless and efficient way for users to engage in online auctions. At its core, the system enables users to register accounts, list items for auction, place bids on items of interest, and facilitate secure payment transactions. Leveraging cutting-edge technologies such as Node.js for scalable and high-performance server-side scripting, and EJS for generating dynamic HTML content, the system embodies the convergence of modern web development principles and auction industry best practices. Whether users are seasoned auction enthusiasts or newcomers exploring the thrill of online bidding, the system caters to a diverse audience, providing an accessible and engaging platform for buying and selling a wide range of goods. With its user-centric design and robust functionality, the online auctioning system stands as a testament to the power of technology in transforming traditional auction processes into dynamic and immersive digital experiences.

# EXPLANATION OF CONTENTS

## ABSTRACT

The online auctioning system represents a comprehensive and sophisticated platform tailored to meet the evolving needs of modern buyers and sellers in the digital marketplace. At its core, the system offers a seamless and intuitive interface for users to engage in dynamic auction experiences. From the initial registration process to the finalization of transactions, the platform streamlines every aspect of the auction process, fostering a secure and efficient environment for buying and selling goods. Leveraging the power of Node.js for robust server-side scripting and EJS for dynamic content generation, the system boasts a scalable and responsive architecture capable of handling a high volume of concurrent users and transactions. Key features include user account management, item listing and categorization, real-time bidding capabilities, and secure payment processing. With its user-centric design and emphasis on accessibility, the system caters to a diverse audience, from seasoned auction veterans to first-time bidders, offering an immersive and engaging experience that transcends traditional auction formats. By harnessing the latest technologies and best practices in web development, the online auctioning system represents a paradigm shift in the way auctions are conducted, empowering users to navigate the digital marketplace with confidence and ease.

# CHAPTER 1

## BACKGROUND

The transition from traditional in-person auctions to online platforms marks a significant evolution in the auction industry, driven by technological advancements and changing consumer behaviors. While traditional auctions were constrained by geographical limitations and restricted audience reach, online auctions break down these barriers, enabling participants from around the world to engage in bidding processes remotely. However, this shift has also highlighted the challenges and limitations of traditional methods, including concerns regarding security, authenticity, and accessibility. As a response to these challenges, the development of online auctioning systems has gained momentum, offering a more accessible, efficient, and secure platform for buying and selling goods in the digital marketplace.

### Existing systems

Existing auctioning systems occur a diverse array of platforms from tradition auctioning systems held in physical locations to modern auction websites available via browsers. They have served as intermediates between buyers and sellers, fostering exchange of products and services through competitive bidding process.

#### Advantages

* Traditional auctions have an increased level of transparency
* There’s global participation that potentially reaches a larger audience.
* Facilitate the general commerce.

#### Disadvantages

* Potential Fraud and breaches.
* Competition among bidders can drive price higher affecting perceived value of the product.

### Proposed systems

The proposed system aims to utilize the advantages of the existing system while addressing their limitations through newer robust functionalities.

Equipped with UpToDate security features, user friendly interfaces and easy accessibility, the proposed system looks forward to creating an engaging marketplace for buyers and sellers alike.

## OBJECTIVES

The objectives of the online auctioning system are multi-faceted, aiming to enhance user experience, streamline auction processes, and increase participation and engagement. By providing users with intuitive interfaces, seamless transactional flows, and robust security measures, the system seeks to improve overall satisfaction and trust in the auction experience. Additionally, the system aims to enhance efficiency in auction transactions, reducing the time and effort required for item listing, bidding, and payment processing. Ultimately, the primary objective is to create a dynamic and engaging platform that empowers users to buy and sell goods with confidence and convenience.

## PURPOSE, SCOPE, APPLICABILITY

### PURPOSE

The purpose of the online auctioning system is to address the challenges and limitations of traditional auction methods while fulfilling unmet needs in the auction marketplace. By leveraging technology to provide a more accessible, efficient, and secure platform, the system aims to overcome geographical constraints, increase market reach, and foster transparency and trust in auction transactions. Furthermore, the system seeks to enhance user experience by offering intuitive interfaces, real-time bidding capabilities, and secure payment mechanisms. Ultimately, the purpose of the system is to revolutionize the way auctions are conducted, empowering users to navigate the digital marketplace with confidence and ease.

### SCOPE

The scope of the online auctioning system encompasses a range of functionalities and features designed to facilitate a seamless and efficient auction experience. Key functionalities include:

* user registration and authentication,
* item listing and categorization,
* real-time bidding capabilities,
* and secure payment processing.
* Additionally, administrative tools are provided to manage user accounts,
* monitor auction activities,
* and ensure compliance with regulations and policies.

While the system focuses on core auction-related activities, certain features such as advanced analytics and third-party integrations are explicitly excluded from the scope to maintain clarity and focus on primary objectives.

### APPLICABILITY

The online auctioning system holds potential applications across various industries and sectors, including e-commerce, collectibles, art, real estate, and automotive. Its versatility and scalability make it suitable for a wide range of goods and services, from consumer products to high-value assets. The primary audience includes buyers, sellers, administrators, and platform operators, each benefiting from the system's user-friendly interfaces, robust security measures, and efficient transactional flows. By facilitating transparent and efficient auction transactions, the system has the potential to drive innovation, increase market reach, and promote trust and credibility in the auction industry, ultimately shaping the future of online commerce.

# CHAPTER 2: SURVEY OF TECHNOLOGIES

## Overview

In the development of the auctioning system a myriad of technologies was employed, each performing distinct tasks but working collaboratively with one another to form the major building blocks of the system.  
The selection of tools and frameworks that supported the development included both back-end and front-end technologies as well as tools for testing, debugging, version control and deployment.

## Rationale

The technologies I chose were selected based on features that are suitable support the system’s functionality.

I looked into security, the scalability, performance and my experience with these technologies to ensure effectiveness of the system.

## Roles

Each technology plays a vital role in the whole system architecture that contributes to user experience and performance.

* Front-end technologies handled the interface design being rendered to users and overall user interaction.
* Back-end technologies manages data storage, server-side processing and communication with clients.

## Programming languages and data stores

* JavaScript serves as the main programming language for both back-end and front-end providing seamless communication with different modules of the system.

Node.js based on JavaScript made it possible to develop front-end and back-end using the same language stack.

* MongoDB serves as our centralized database. It’s scalability capabilities makes it a good choice for handling ever increasing user data overtime.

## Frameworks and Libraries

* Express.js, a framework built for Node.js is used for backend development providing a robust set of features for building web applications and APIs.
* Tailwind CSS – A rapidly growing CSS framework is employed for front-end design providing responsive components and incredible user experience.
* Embedded JavaScript (Ejs) – Responsible for rendering the contents served to a user after initiating a request. It works hand in hand with express.js and tailwind CSS to render visually impressive websites.
* Mongoose – A Node.js and MongoDB ORM (Object Relational Mapping) that is responsible for interactions and CRUD (Create, Read, Update, Delete) operations on our database.
* bcrypt – A framework responsible for encrypting passwords and other sensitive information in our database.
* express-session – Responsible for user authentication, serializing and deserializing users into their activity sessions.
* Body-parser – Allows seamless passing user generated data from the browser to the server-side code for processing.
* uuid – creates unique ids for objects in the database such that records have no duplicates and do not conflict.

## Tools

* VSCode – An efficient and customizable IDE offering a wide range of extension to leverage.
* Nodemon – A Node.js library responsible for monitoring changes to our project folder and restarting our server to accommodate changes
* Postman – An API testing software for testing the functionality of API, performance and speeds.
* REST Client – An API testing extension integrated within VSCode.
* ESLint – A linter responsible for exposing errors in JavaScript code.
* Git – A version control software for tracking changes and versions of the development
* Github – A GUI and web based version control software that works hand in hand with Git to provide hosting to our source code.

# CHAPTER 3: REQUIREMENTS AND ANALYSIS

## 3.1 PROBLEM DEFINITION

The problem definition for the proposed online auctioning system revolves around addressing the limitations and challenges inherent in traditional auction methods while leveraging the opportunities presented by digital technology. Traditional auction processes often suffer from geographical constraints, limited reach, lack of transparency, and security concerns, which can hinder the efficiency and fairness of the bidding process. Additionally, manual handling of auction logistics can lead to inefficiencies and errors, impacting both buyers and sellers. The proposed system aims to overcome these challenges by providing a secure, transparent, and accessible platform for conducting online auctions. By migrating the auction process to a digital environment, the system seeks to expand market reach, enhance transparency, streamline transaction processes, and improve overall user experience for buyers and sellers alike. Through real-time bidding, secure payment processing, comprehensive item listings, and robust security measures, the proposed system aims to revolutionize the way auctions are conducted, making it easier, safer, and more efficient for participants to buy and sell goods and services online.

## 3.2 REQUIREMENT SPECIFICATION

The requirements specifications for the proposed online auctioning system outline both functional and non-functional requirements essential for its development and implementation. Functionally, the system must allow users to securely register accounts, list items for auction with detailed descriptions, facilitate real-time bidding, process payments securely, and provide administrators with tools for user and item management, as well as reporting and analytics. Non-functionally, the system must prioritize security through encryption, authentication, and data privacy measures, while also ensuring scalability to handle varying levels of user traffic, usability across different devices, reliability with minimal downtime, compatibility with modern web standards, compliance with relevant regulations, and optimal performance with fast loading times and efficient processing. By adhering to these comprehensive requirements, the proposed system aims to deliver a secure, reliable, and user-friendly platform for conducting online auctions effectively and efficiently.

### 3.2.1 HARDWARE REQUIREMENTS

The hardware requirements for the proposed online auctioning system include a reliable server infrastructure capable of hosting the web application and supporting its expected user traffic. This infrastructure should comprise high-performance servers equipped with sufficient processing power, memory, and storage capacity to handle concurrent user interactions, data storage, and transaction processing. Additionally, redundant network connectivity and backup power sources are essential to ensure system availability and resilience against potential failures or disruptions. Furthermore, the server infrastructure should be scalable to accommodate future growth in user demand and data volume. Alongside the server infrastructure, clients accessing the system should have compatible devices with internet connectivity, including desktop computers, laptops, tablets, and smartphones, to ensure accessibility across a diverse range of user demographics. Overall, by investing in robust server infrastructure and supporting client devices, the online auctioning system can deliver a seamless and responsive user experience while meeting the demands of its user base.

### 3.2.2 SOFTWARE REQUIREMENTS

The software requirements for the proposed online auctioning system encompass a comprehensive set of tools and technologies to support its development, deployment, and operation. At the server-side, the system requires a robust operating system such as Linux or Windows Server to host the web application. The system relies on a programming language like JavaScript, supported by a runtime environment such as Node.js, and a web application framework like Express.js to facilitate the development of server-side logic and APIs. Additionally, the system utilizes a database management system such as MongoDB to store and manage data, ensuring reliability, scalability, and data integrity. At the client-side, the system requires modern web browsers with support for HTML5, CSS3, and JavaScript to render and interact with the user interface seamlessly. Development and collaboration tools such as Visual Studio Code, Git for version control through the development process and ensure code quality and consistency. By leveraging these software requirements, the online auctioning system can be developed, deployed, and maintained effectively, providing a secure, reliable, and user-friendly platform for conducting online auctions.

### 3.2.3 FUNCTIONAL REQUIREMENTS

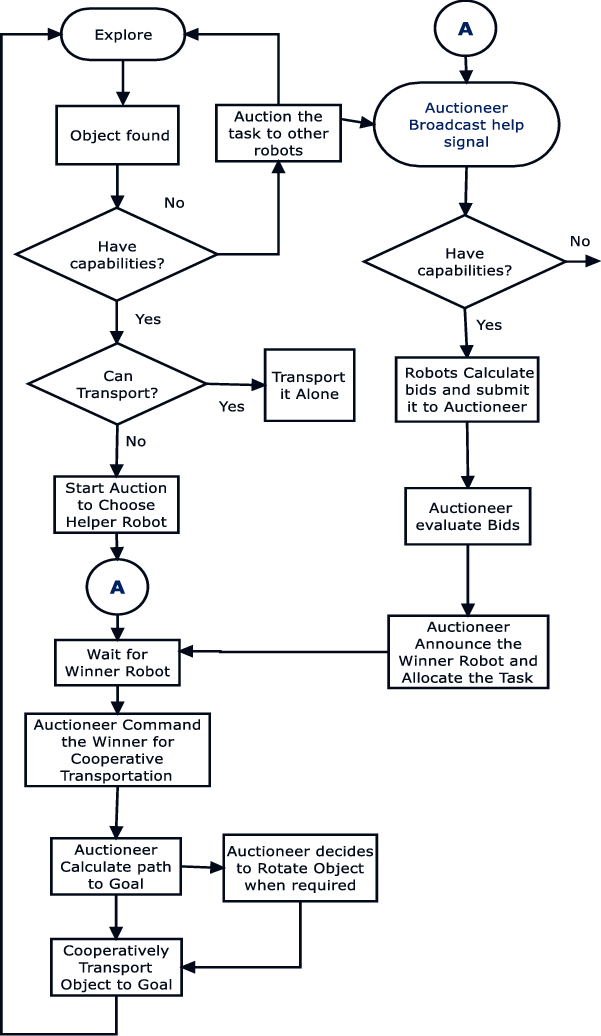
### 3.2.4 NON-FUNCTIONAL REQUIREMENTS

## 3.3 PRELIMINARY PRODUCT DESCRIPTION

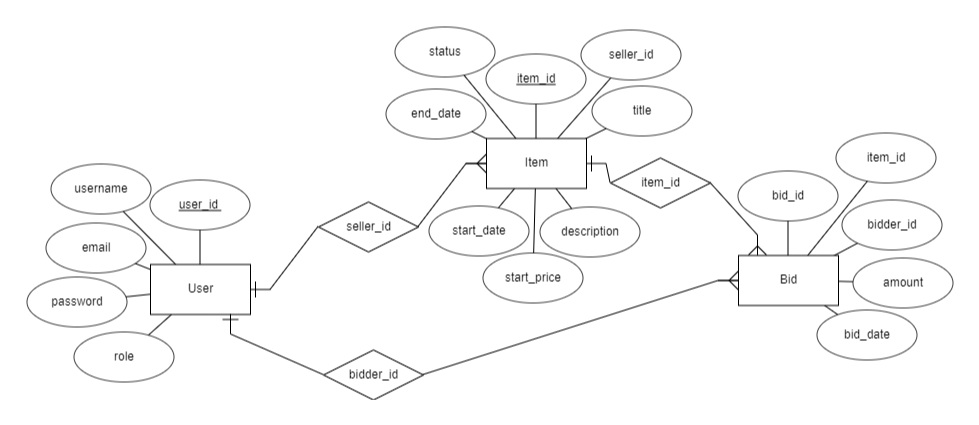
The preliminary product description for the proposed online auctioning system outlines its core features, functionalities, and intended user experience. At its core, the system provides a secure and user-friendly platform for conducting online auctions, facilitating seamless interactions between buyers and sellers. Users can register accounts securely, list items for auction with detailed descriptions and images, and participate in real-time bidding on listed items. The system supports secure payment processing, ensuring that successful transactions are completed efficiently and securely. Administrators have access to comprehensive tools for managing user accounts, monitoring listed items, and generating reports and analytics to track system performance. Additionally, the system features a responsive and intuitive user interface, accessible across different devices and screen sizes, ensuring a seamless user experience for buyers and sellers alike. Overall, the preliminary product description sets the stage for the development of a robust, reliable, and user-centric online auctioning platform, poised to revolutionize the way auctions are conducted in the digital age.

## 3.4 CONCEPTUAL MODELS

### SYSTEM FLOWCHART



### ENTITY RELATIONSHIP DIAGRAMS



# CHAPTER 4: SYSTEM DESIGN

## 4.1 BASIC MODULES

• The online auctioning system comprises several basic modules or components that collectively enable the platform's functionality. These modules typically include user management, item management, bidding, payment processing, and administrative tools. Each module serves a distinct purpose and encapsulates related functionality, allowing for easier maintenance, scalability, and extensibility of the system.

#### Login

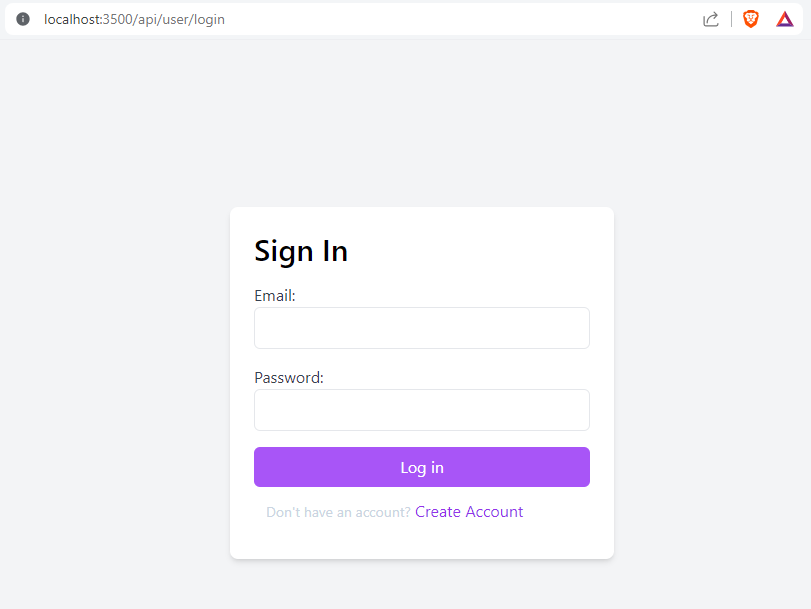


Figure 1 Login



Figure 2 login code

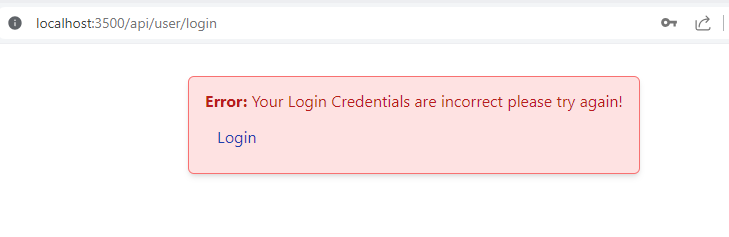
**

Figure 3 invalid login credentials

#### Registration

Figure 4 signup code

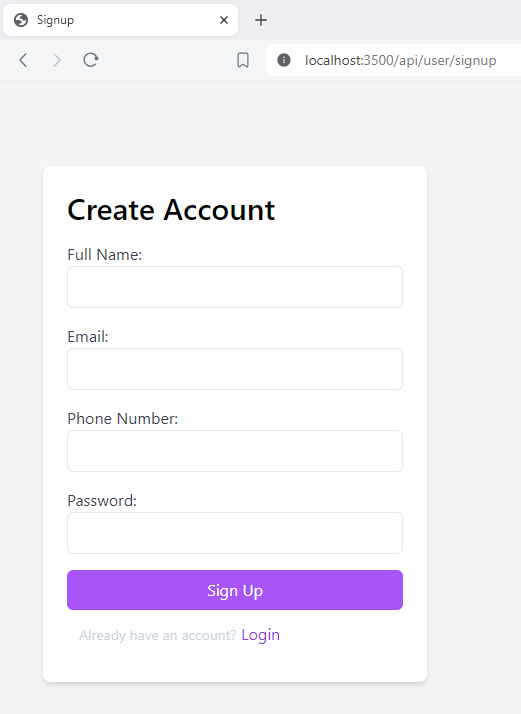


Figure 5 signup page

#### Dashboard

Figure 6 User Dashboard

#### Product upload

Figure 7 Product info upload

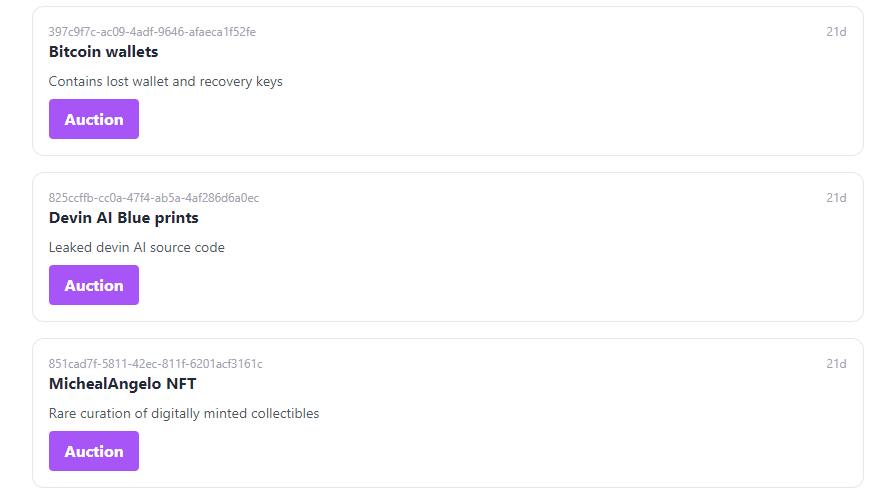


Figure 8 Product list

#### Auctioning

Figure 9 Listing a product for auctioning

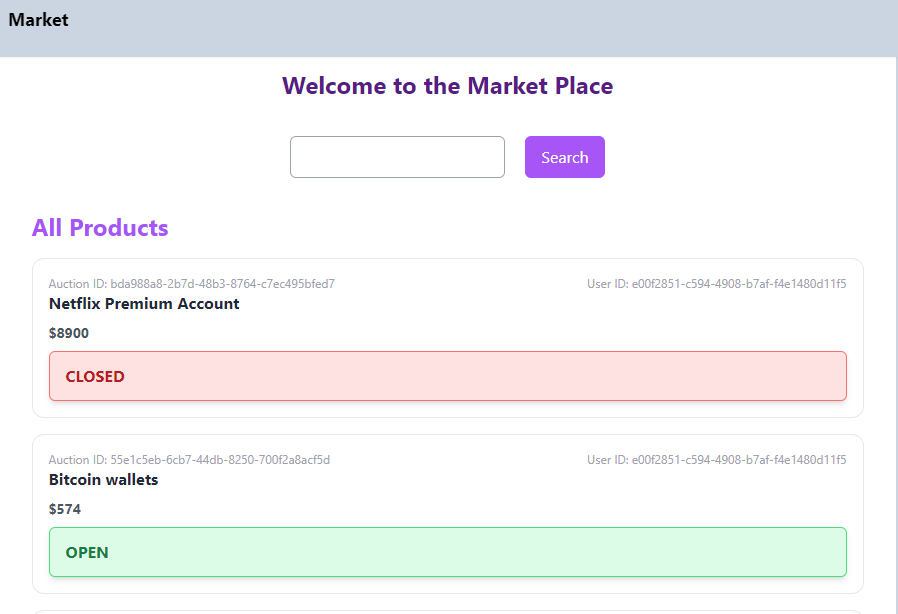


Figure 10 Auctioning market place

#### Bidding

Figure 11 Bidding process

#### Transactions

Figure 12 seller upload auctioned product

Figure 13 Transactions page

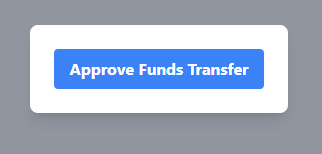


Figure 14 Buyer makes funds transactions

#### Account

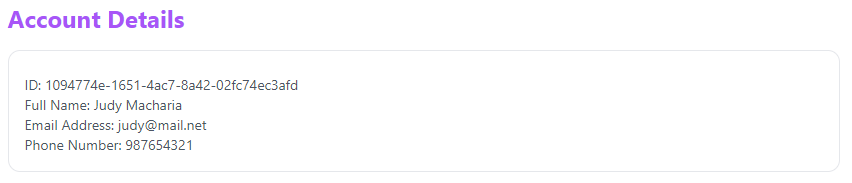


Figure 15 Account details

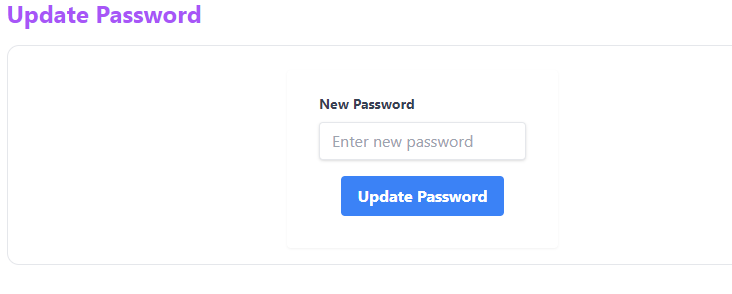


Figure 16 Update password interface



Figure 17 Update password code

## 4.2 DATA DESIGN

### 4.2.1 SCHEMA DESIGN

• The data design of the online auctioning system involves structuring the database schema to efficiently store and manage data. This includes defining tables, fields, relationships, and constraints to ensure data integrity and optimize query performance. By carefully designing the schema, the system can effectively organize and retrieve information related to user profiles, auction items, bids, transactions, and other entities.

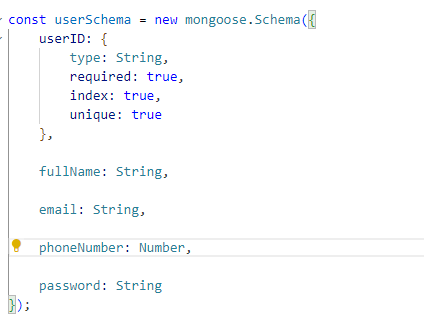


Figure 18 User table Schema

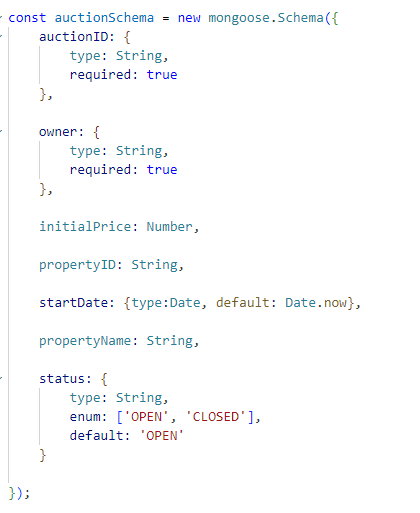


Figure 19 Auctions table schema

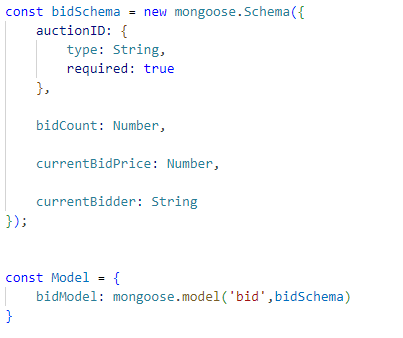


Figure 20 Bidding table schema

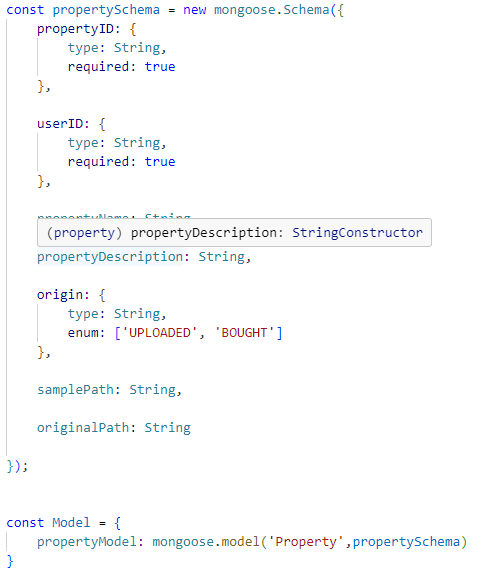


Figure 21 Property table schema

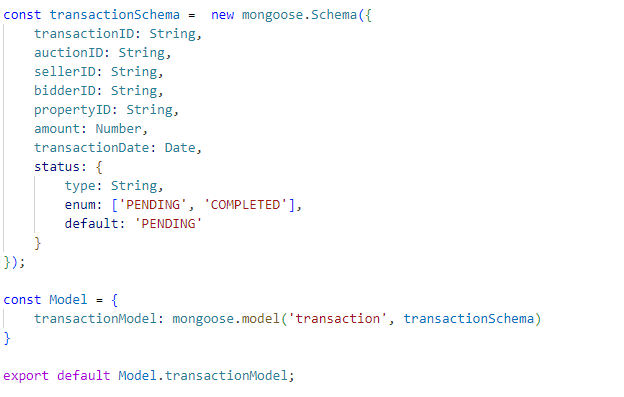


Figure 22 Transactions table schema

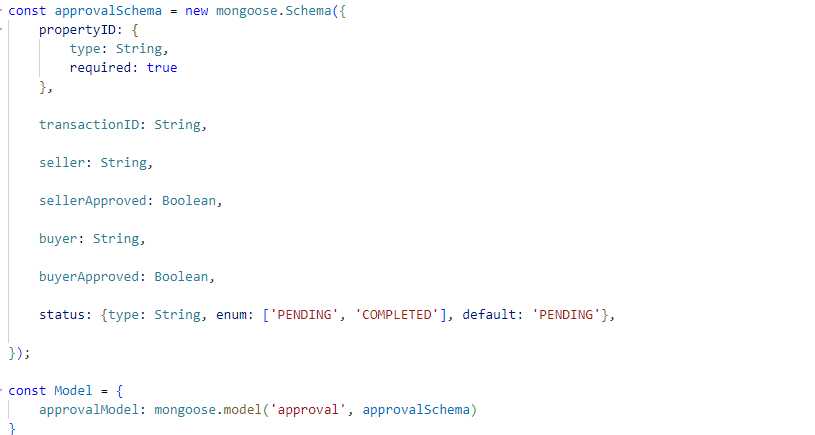


Figure 23 Approvals table schema

### 4.2.2 DATA INTEGRITY AND CONSTRAINTS

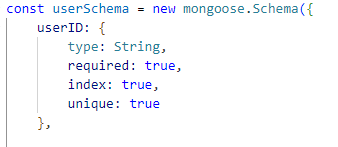
• Data integrity rules and constraints play a crucial role in maintaining the accuracy and consistency of the system's data. These constraints enforce rules such as referential integrity, uniqueness, and data validity, preventing inconsistencies or errors in the database. By implementing constraints such as primary keys, foreign keys, unique constraints, and check constraints, the system can uphold data integrity and reliability.

Figure 24 Integrity constraint, unique key and validation for field UserID

## 4.3 PROCEDURAL DESIGN

### 4.3.1 LOGIC DIAGRAMS

• Logic diagrams provide a visual representation of the logical flow of processes within the online auctioning system. These diagrams illustrate the sequence of steps involved in key processes such as user registration, item listing, bidding, payment processing, and administrative tasks. By mapping out the workflow, logic diagrams help clarify the system's behavior and facilitate communication among stakeholders.

### 4.3.2 DATA STRUCTURES

• Data structures define how data is organized and manipulated within the system. By selecting appropriate data structures such as arrays, lists, queues, stacks, trees, and hash tables, the system can efficiently store, retrieve, and process information. These data structures play a crucial role in optimizing the performance and scalability of the system, ensuring efficient handling of large volumes of data.

### 4.3.3 ALGORITHMS DESIGN

• Algorithms are the underlying logic that drives the functionality and operations of the online auctioning system. These algorithms define the step-by-step procedures for tasks such as user authentication, item sorting and filtering, bid validation, payment processing, and system monitoring. By designing efficient algorithms, the system can perform tasks quickly and accurately, delivering a seamless user experience.

### 4.3.4 USER INTERFACE DESIGN

• The user interface (UI) design of the online auctioning system focuses on creating intuitive and visually appealing interfaces for users. Wireframes or mockups are used to visualize the layout, navigation, and interaction flow of the system. By incorporating responsive design principles, the UI design ensures that the system is accessible and usable across different devices and screen sizes, enhancing the overall user experience.

### 4.3.5 SECURITY ISSUES

• Security is paramount in the design of the online auctioning system to protect user data and prevent unauthorized access or malicious activities. The system must implement robust security measures such as authentication, authorization, encryption, input validation, and error handling. By identifying and mitigating potential security threats and vulnerabilities, the system can safeguard sensitive information and ensure the integrity and confidentiality of user data.

### 4.3.6 TEST CASE DESIGN

• Test cases are essential for validating the functionality and performance of the online auctioning system. These test cases cover various scenarios and procedures to ensure that each module or component of the system behaves as expected. Test cases include positive scenarios to verify correct system behavior as well as negative scenarios to test error handling and edge cases. By thoroughly testing the system, issues can be identified and addressed early in the development process, ensuring a robust and reliable product.

# CHAPTER 5: IMPLEMENTATION AND TESTING

## 5.1 IMPLEMENTATION APPROACHES

• The implementation approach for the online auctioning system followed an agile methodology, enabling iterative development and continuous feedback. Agile principles such as collaboration, adaptability, and incremental delivery were emphasized throughout the development process. Regular sprint cycles allowed for frequent releases and rapid response to changing requirements, ensuring that the system remained aligned with stakeholder expectations. By adopting an agile approach, the development team was able to prioritize features, address issues promptly, and deliver a high-quality product within the specified timeframe.

## 5.2 CODING DETAILS AND CODE EFFICIENCY

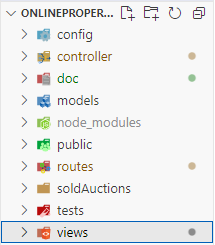
• The online auctioning system was implemented using JavaScript for both front-end and back-end development, leveraging the Node.js runtime environment and Express.js framework. The codebase followed modular design principles, with each module encapsulating specific functionality and promoting code reusability. Design patterns such as MVC (Model-View-Controller) were employed to separate concerns and maintain a clean and organized codebase. Additionally, code efficiency was prioritized through performance optimization techniques such as caching, lazy loading, and asynchronous programming, ensuring that the system remained responsive and scalable under varying loads.

Figure 25 MVC approach

## 5.3 TESTING APPROACHES

• The testing approach for the online auctioning system combined automated and manual testing techniques to ensure thorough coverage and reliable results. Manual testing was performed for exploratory testing, usability testing, and user acceptance testing, allowing for real-world validation of the system's functionality and user experience.

### 5.3.1 UNIT TESTIING

• Unit testing played a crucial role in validating the functionality and correctness of individual modules or components within the online auctioning system. Test cases were written to cover both positive and negative scenarios, ensuring that each function or method behaved as expected under various conditions. Test automation tools such as Jest and Mocha were used to automate the execution of unit tests, allowing for quick feedback and regression testing during development. By incorporating unit testing into the development process, the team was able to detect and fix bugs early, improving code quality and reducing the risk of defects in the final product.

Figure 26 Unit testing

### 5.3.2 INTEGRATION TESTING

• Integration testing focused on verifying the interaction and compatibility of different modules or components within the online auctioning system. Test scenarios were designed to simulate real-world usage scenarios, covering end-to-end workflows such as user registration, item listing, bidding, and payment processing. Test environment setup included deploying the system to staging or testing servers to replicate production-like conditions. Challenges such as dependencies between modules and data synchronization were addressed through thorough test planning and execution. Integration testing helped ensure that the system functioned as expected when all components were integrated, providing confidence in its overall reliability and performance.



Figure 27 Integration testing

## 5.4 MODIFICATIONS AND IMPROVEMENTS

• Throughout the implementation and testing phases, several modifications and improvements were made to the online auctioning system based on feedback from testing activities and user evaluations. These modifications included enhancements to user interface design, optimization of database queries for improved performance, and refinement of business logic to better align with user requirements. Lessons learned from testing activities were incorporated into future development iterations, guiding decisions on feature prioritization and technical refinements. Continuous improvement efforts focused on delivering a robust and user-friendly platform that met the evolving needs of its users and stakeholders.

## 5.5 ENVIRONMENT SETUP

* Download the appropriate installer for your operating system (Windows, macOS, or Linux) by selecting the "LTS" (Long Term Support) version.
* Run the downloaded installer and follow the on-screen instructions to complete the installation.
* Verify that Node.js and npm (Node Package Manager) are installed correctly by opening a terminal or command prompt and running the following commands:
  + **node -v**
  + **npm –v**

#### Creating a New Express.js Project

* Open a terminal or command prompt window.

Since we are using the same directorate for express js we are going to create another directorate in this directorate where our express js will be installed.

* Create a directory for your Express.js project

**mkdir my-express-app**

* Navigate to the project directory:

**cd my-express-app**

* Initialize a new Node.js project:

**npm init –y**

* Install Express.js as a dependency:

**npm install express**

* Write your Express.js application code in a new JavaScript file (e.g., app.js).
* Start the Express.js server by running:

**node app.js**

# CHAPTER 6: CONCLUSIONS

## 6.1 CONCLUSIONS

• Throughout the development journey of the online auctioning system, our team has achieved significant milestones and addressed various challenges to deliver a robust and user-friendly platform. The system successfully meets its primary objectives of enhancing user experience, streamlining auction processes, and providing a secure and efficient marketplace for buyers and sellers alike. With features such as real-time bidding, secure payment processing, and intuitive user interfaces, the system offers a compelling solution to the limitations of traditional auction methods. We extend our heartfelt appreciation to all team members, stakeholders, and contributors for their dedication and collaboration in bringing this project to fruition.

## 6.2 LIMITATIONS OF THE SYSTEM

• While the online auctioning system boasts many strengths, it is not without its limitations. Technical constraints, such as occasional scalability issues during peak usage periods and compatibility challenges with older browsers, may impact the system's performance and user experience. Additionally, usability limitations, such as complex navigation flows and accessibility barriers for users with disabilities, represent areas for improvement in future iterations. Furthermore, compliance with evolving data privacy regulations and industry standards presents ongoing challenges that require continuous attention and adaptation.

## 6.3 FUTURE DIRECTIONS

• Looking ahead, there are several exciting opportunities for further development and enhancement of the online auctioning system. Future iterations could focus on optimizing system scalability and performance, refining user interfaces to improve usability and accessibility, and implementing advanced features such as machine learning algorithms for personalized recommendations and fraud detection. Additionally, exploring partnerships with third-party vendors and expanding into new markets could help broaden the system's reach and impact. By prioritizing ongoing support, maintenance, and innovation, we can ensure that the online auctioning system continues to evolve and thrive in the dynamic landscape of e-commerce and auctioning.

# References:

1. Anderson, S., de Palma, A., & Thisse, J.-F. (1992). Discrete Choice Theory of Product Differentiation. Cambridge, MA: The MIT Press.
2. Krishnamurthy, S. (2002). The Multibidder Dutch Auction. Journal of Economic Theory, 105(2), 444-456.
3. McAfee, R. P., & Reny, P. J. (1992). Correlated Information and Mechanism Design. Econometrica, 60(2), 395-421.
4. Rothkopf, M. H. (2007). Thirteen Classic Papers on Auctions. Decision Analysis, 4(2), 97-99.