**PROJECT REPORT OF EXPLORATORY PROJECT**

**ON**

BidCommerce

**submitted in partial fulfilment of the requirements for the award of degree of**

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by: Supervised By:**

**Anuj Garg Dr. Ajay Kumar**

**2110991644 Assistant Professor**

**Chitkara University**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CHITKARA UNIVERSITY**

**CHANDIGARH-PATIALA NATIONAL HIGHWAY, RAJPURA, PUNJAB, INDIA**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Topic** | **Page No** |
| **1** | **Abstract** | **1** |
| **2** | **Introduction** | **2** |
| **3** | **Problem Statement** | **3** |
| **4** | **Objective** | **4** |
| **5** | **Methodology** | **5** |
| **6** | **Tools and Technologies** | **6** |
| **7** | **Project Plan** | **8** |
| **8** | **Conclusion** | **10** |
| **9** | **References** | **11** |

**DECLARATION**

I hereby certify that the work which is being presented in the project report entitled

“**BidCommerce**” in partial fulfilment of requirement for the award of the degree of Bachelor of Engineering (Computer

Science and Engineering) submitted in the department of Computer Science and Engineering at Chitkara

University Institute of Engineering and Technology, Chitkara University, Punjab, India, is an authentic

record of my own work carried out under the supervision of **Dr. Ajay Kumar**.The matter presented in

this project report has not been submitted in any other university/institute for the award of any degree.

Place: Rajpura Anuj Garg

Date: 12-12-2024 2110991644

This is to certify that the above statement made by the candidate is correct to the best of my knowledge

and belief.

Dr. Ajay kumar

Assistant Professor

Department of Computer Science and Engineering

Chitkara University Institute of Engineering and Technology,

Chitkara University, Punjab, India

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to my project guide Dr. Ajay Kumar, whose guidance and support were instrumental in the completion of this project. I would also like to thank my family and friends for their encouragement and collaboration. Lastly, I acknowledge the help of online resources and tutorials that facilitated the development of the BidCommerce Website.

**Abstract**

The **BidCommerce** Website, developed using the MERN (MongoDB, Express, React, Node.js) stack, provides a comprehensive platform for users to explore and bid on products dynamically. The platform includes features such as secure user authentication for login and signup, personalized user experiences, and dynamic bidding functionalities.

Users can list products for bidding, manage orders, and track their purchases in real-time. The system enables competitive bidding, ensuring users can participate in auctions seamlessly. Additionally, it offers flexibility for users to modify their bids and provides a structured approach to handling e-commerce transactions.

With an intuitive user interface and robust functionality, the BidCommerce Website empowers users to experience a unique and dynamic e-commerce platform. Leveraging the MERN stack, the website ensures scalability, security, and a smooth user experience for buyers and sellers.

**Introduction**

The BidCommerce Website is an e-commerce application developed using the MERN stack, providing a dynamic bidding platform for users. The website allows users to register, log in, and participate in competitive auctions. Sellers can list products with details like pricing and bidding deadlines, while buyers can explore and bid on these products.

The MERN stack provides the necessary infrastructure to create a robust e-commerce platform. MongoDB is used to store user data, product details, and bidding records. Express.js and Node.js handle the backend logic and API development, while React ensures a dynamic and responsive user interface.

This platform offers a seamless user experience, ensuring secure transactions, efficient product management, and dynamic bidding, catering to modern e-commerce needs.

The MERN stack, which consists of MongoDB, Express.js, React, and Node.js, provides a robust and efficient

framework for developing such applications. MongoDB is used as the database to store user information and exercise

data, Express.js facilitates server-side operations, React enables the creation of interactive user interfaces, and Node.

js manages the server-side logic.

By leveraging the MERN stack, developers can create a responsive and feature-rich BidCommerce that helps users

stay organized, motivated, and on track with their buying goals.

**Problem Statement**

In the rapidly growing e-commerce industry, there is a lack of platforms that provide competitive bidding options for products, making it challenging for users to get the best deals. Sellers also face difficulties in managing their product catalogs and engaging potential buyers effectively.

Problem Statement:

"Many e-commerce platforms lack dynamic bidding functionalities, resulting in missed opportunities for both buyers and sellers. Buyers often struggle to secure competitive prices, while sellers lack efficient tools to attract potential bidders. A robust e-commerce solution with an integrated bidding system is needed to bridge this gap, empowering users with seamless transactions and a competitive edge."

### **Objective**

The objective of this project is to develop a feature-rich e-commerce platform with a focus on dynamic bidding functionalities using the MERN stack. The key goals of the project include:

1. **User Authentication**: Secure login and signup to protect user data and personalize the experience.
2. **Product Managemen**t: Enable sellers to list products with detailed descriptions and bidding options.
3. **Bidding System**: Allow buyers to bid dynamically on products, ensuring competitive pricing.
4. **User-Friendly Interface:** Design an intuitive interface to simplify navigation and bidding.
5. **Payment Gateway Integration**: Provide secure payment options for smooth transactions.
6. **Order Management:** Enable users to track orders and bidding histories.
7. **Scalability**: Ensure the platform can handle increasing user demands and product listings.
8. **Real-Time Updates**: Display live bidding updates to keep users informed.
9. **Deployment**: Deploy the application on a scalable and reliable hosting platform.

**Methodology**

Here is a methodology for developing a Bidcommerce using the MERN stack

* **Planning Phase:**
  + Define goals: Dynamic bidding, user authentication, and product management.
  + Identify requirements and design wireframes.
* **Database Design:**
  + MongoDB for user profiles, product details, and bidding records.
  + Define relationships between users, products, and bids.
* **Backend Development (Node.js and Express):**
  + Set up APIs for login, signup, product management, and bidding.
  + Implement secure payment gateway integration.
* **Frontend Development (React):**
  + Create interfaces for product listings, bidding, and checkout.
  + Use real-time updates for bidding information.
* **Testing & Deployment:**
  + Perform unit and integration testing.
  + Deploy on platforms like AWS or Heroku.

**Tools And Technologies**

To develop a BidCommerce using the MERN stack with login, signup,bidding and product management, and removal functionalities, you would typically use the following tools and technologies:

* **Client-Side (Frontend):**
  + React Components: User interface components for login, signup, new addition, and removal functionalities.
  + Redux (Optional): State management for maintaining user data and application state across components.
  + React Router: Client-side routing for navigation between different views and components.
  + Axios: HTTP client for making API requests to the backend server.
* **Server-Side (Backend):**
  + Express.js: Backend framework for handling HTTP requests and routing.
  + Authentication Middleware: Middleware for user authentication using JWT (JSON Web Tokens) during login and signup processes.
  + API Endpoints: RESTful APIs for CRUD operations on various things and user data.
  + Mongoose: ODM (Object Data Modeling) library for MongoDB to interact with the database.
* **Database (MongoDB):**
  + Collections: Separate collections for storing user information
  + Indexes: Indexing key fields to optimize query performance.
  + Relationships: Establishing relationships between user data and items and cart products
* **Authentication Flow:**
  + User initiates login/signup request from the frontend.
  + Backend verifies credentials and generates JWT token upon successful authentication.
  + Token is stored on the client-side for subsequent authenticated requests.
* **Security Considerations:**
  + Implementing secure communication over HTTPS to protect data in transit.
  + Input validation to prevent injection attacks and ensure data integrity.
  + Proper error handling to maintain system stability and security.

### **Project Duration**

Start Date: 20/07/2024  
End Date: 04/12/2024

### Implementation Plan

### Phase 1: Planning and Design (20/07/2024 - 10/08/2024)

* Requirement Analysis: Document detailed requirements for each feature.
* Design: Create wireframes and prototypes for all sections (User Authentication, Management, User Interface, etc.).
* Technology Selection: Finalize the technology stack (React, Node.js, MongoDB, etc.).

#### Phase 2: Development (11/08/2024 - 10/09/2024)

* Frontend Development: Develop the user interface using React, HTML, CSS, and JavaScript.
* Backend Development: Set up the server using Node.js and Express.js. Develop APIs for user authentication, data retrieval.
* Database Integration: Implement MongoDB for storing user data,details.

#### Phase 3: Feature Implementation (11/09/2024 - 01/10/2024)

* User Authentication: Implement secure login and signup functionalities.
* **BidCommerce** Management: Develop features for adding, removing, and product management and bidding.
* User Interface: Enhance the UI to be intuitive and user-friendly.
* Customizable Options: Allow users to customize their bidding plans and preferences.

#### Phase 4: Testing (02/10/2024 - 01/11/2024)

* Unit Testing: Test individual components for functionality.
* Integration Testing: Ensure seamless interaction between frontend and backend components.
* User Acceptance Testing: Validate the system against user requirements and feedback.

#### Phase 5: Deployment (02/11/2024 - 15/11/2024)

* Hosting: Deploy the application on a reliable cloud platform (AWS, Azure, Heroku, etc.).

#### Phase 6: Maintenance and Support (16/11/2024 - 04/12/2024 and ongoing)

* Regular Updates: Provide regular updates and new features based on user feedback.
* Bug Fixes: Address any issues or bugs that arise.
* User Support: Offer robust customer support to assist users with any questions or problems.

By adhering to this project plan, the products and bidding aims to deliver a robust and user-friendly platform that supports individuals in achieving their buying goals efficiently and effectively.

**Major findings and outcomes**

* Seamless integration of bidding functionalities for dynamic pricing.
* A user-friendly interface for both buyers and sellers.
* Secure transactions with robust authentication and payment gateways.
* Real-time updates to improve user engagement.
* Scalable and responsive design to accommodate growing user demands.

### **Conclusion**

The **BidCommerce Website** project successfully delivers a robust and scalable e-commerce platform with dynamic bidding functionalities. By addressing gaps in existing e-commerce solutions, the website empowers users to participate in competitive auctions while ensuring a seamless experience for buyers and sellers.

The project incorporates modern web technologies like React.js, Node.js, Express.js, and MongoDB, providing a comprehensive tool to manage products, bids, and transactions efficiently. Continuous improvements and user feedback will help evolve the platform to meet future e-commerce challenges and user expectations.

**REFERENCES**

For the References section, I list the online resources, libraries, and documentation that helped me complete my project.

**Official documentation**

React Documentation: <https://reactjs.org/docs/getting-started.html>

Node.js Documentation: <https://nodejs.org/en/docs/>

MongoDB Documentation: <https://www.mongodb.com/docs/>

Express.js documentation: <https://expressjs.com/en/guide/routing.html>

Redux documentation: <https://redux.js.org/usage/>

**Libraries and packages**

"react": <https://www.npmjs.com/package/react>

"express": https://www.npmjs.com/package/express

"node.js" : https://www.npmjs.com/package/node

**Other resources**

Stack Overflow: [https://stackoverflow.com .](https://stackoverflow.com/)

GitHub repositories related to file upload and processing.