



K.R. MANGALAM UNIVERSITY

THE COMPLETE WORLD OF EDUCATION

Recognised under the section 2 (f) of the UGC Act 1956



Empowering the Youth; Empowering the Nation





K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

E-2 Market

Second Year Project Synopsis Submitted by

ROLL	NAME
2301010174	Anurag Nidhi

Industry Mentor: Arun Kumar
Faculty Mentor: Ms Kriti

Project Overview

E-2 Market is an online platform designed to facilitate the buying and selling of used products in a secure, user-friendly, and transparent manner. The platform aims to decrease the gap between sellers looking to declutter and buyers searching for affordable, second-hand alternatives

The platform is built using the MERN stack (MongoDB, Express.js, React.js, Node.js) to ensure scalability, responsiveness, and efficient performance. Key features include advanced search filters, personalized recommendations, a rating and review system, and secure payment options. These functionalities enhance user experience while ensuring trust and security in transactions.



About the Problem

- 1. Limited Access to Local Marketplaces:** Local buyers and sellers often struggle to find relevant products or reach many potential customers in their area.
- 2. Limited Market Reach:** Smaller sellers often struggle to reach a larger customer base or have trouble navigating big, cluttered platforms.
- 3. Improved User Experience:** Implement intuitive features like product filters, direct messaging, and payment integration for ease of use.



Problem Statement

Defining the Problem

Many online platforms for buying and selling used products, Many marketplaces, have issues like:

- **Lack of security** – Users face fraud and scams.
- **Poor search options** – It's hard to find the right products.
- **Limited trust** – No proper user verification, leading to fake listings.

Why Is This Important?

Second-hand shopping is growing, but people need a **safe and reliable** platform. Existing websites don't always provide trust and ease of use. Our platform, **E-2 Market**, will solve these problems.

Expected Impact of E-2 Market

- **More Trust:** Secure payments and verified users reduce scams.
- **Better Experience:** Easy navigation and smart search filters.
- **Local Convenience:** Connects nearby buyers and sellers.
- **Transparency:** Ratings and reviews help users make informed choices.



Objectives

1. To design and develop a user-friendly online marketplace for buying and selling used products, ensuring a seamless and intuitive experience for both buyers and sellers.
2. To integrate advanced search filters and personalized recommendations into the platform, allowing users to easily find products that match their preferences and needs.
3. To implement a reliable rating and review system that builds trust between users and promotes transparency in product listings.
4. To build a scalable platform using the MERN stack (MongoDB, Express.js, React.js, Node.js) that can handle growing user traffic and expanding product listings.



Tools, and Techniques

Tools/Technologies Used

1. PROGRAMMING LANGUAGE: JAVASCRIPT (React.js & Node.js)

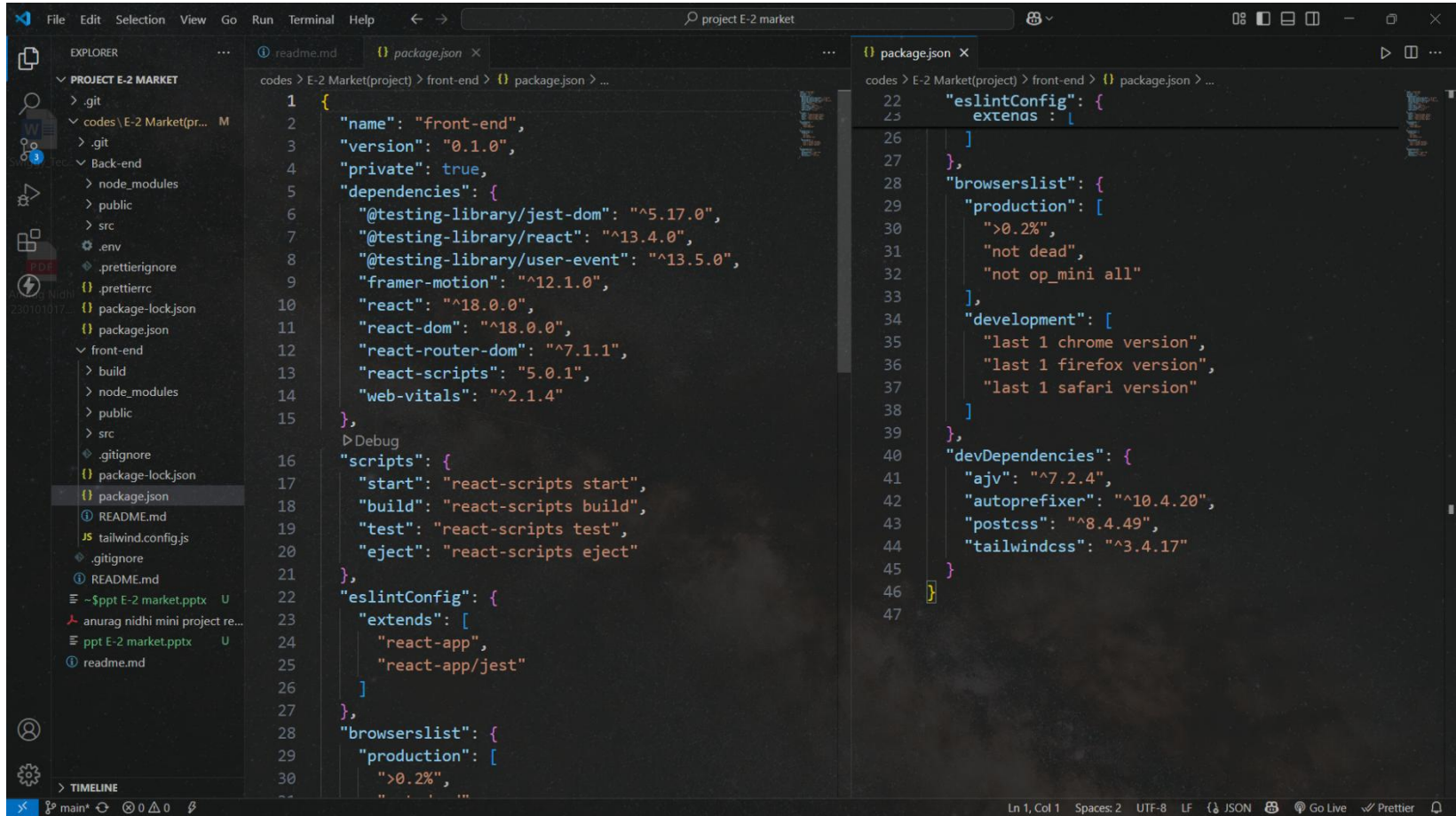
JavaScript is the primary programming language used for the front-end and back-end development of the platform. React.js is used for building a dynamic and responsive user interface, while Node.js is employed for the back-end to handle requests, databases, and server-side logic. JavaScript is essential due to its versatility and wide usage in modern web development.

2. DATABASE: MONGODB

MongoDB is a NoSQL database used to store and manage product listings, user information, and transaction details. As a document-based database, it offers scalability and flexibility, making it an ideal choice for handling large volumes of unstructured data, which is a common requirement for e-commerce platforms like E-2 Market.



Tools, and Techniques



The screenshot displays the Visual Studio Code interface with a project named "project E-2 market". The Explorer sidebar on the left shows the project structure, including a "front-end" directory. The main editor area is split into two panes, both displaying the "package.json" file for the "front-end" directory.

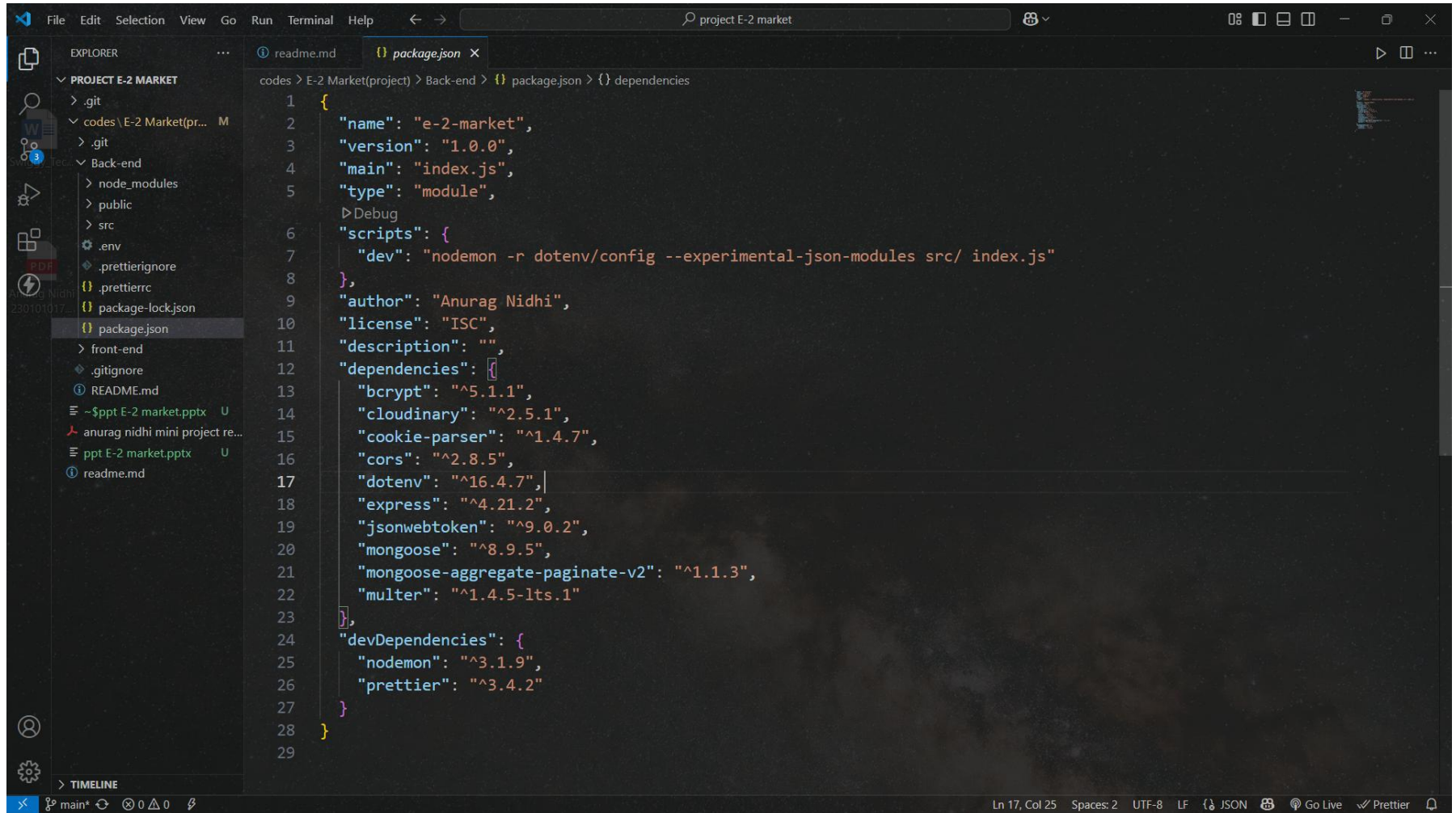
The left pane shows the full "package.json" file:

```
1 {
2   "name": "front-end",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@testing-library/jest-dom": "^5.17.0",
7     "@testing-library/react": "^13.4.0",
8     "@testing-library/user-event": "^13.5.0",
9     "framer-motion": "^12.1.0",
10    "react": "^18.0.0",
11    "react-dom": "^18.0.0",
12    "react-router-dom": "^7.1.1",
13    "react-scripts": "5.0.1",
14    "web-vitals": "^2.1.4"
15  },
16  "scripts": {
17    "start": "react-scripts start",
18    "build": "react-scripts build",
19    "test": "react-scripts test",
20    "eject": "react-scripts eject"
21  },
22  "eslintConfig": {
23    "extends": [
24      "react-app",
25      "react-app/jest"
26    ]
27  },
28  "browserslist": {
29    "production": [
30      ">0.2%",
31      "not dead",
32      "not op_mini all"
33    ],
34    "development": [
35      "last 1 chrome version",
36      "last 1 firefox version",
37      "last 1 safari version"
38    ]
39  },
40  "devDependencies": {
41    "ajv": "^7.2.4",
42    "autoprefixer": "^10.4.20",
43    "postcss": "^8.4.49",
44    "tailwindcss": "^3.4.17"
45  }
46 }
```

The right pane shows a snippet of the "package.json" file, focusing on the "eslintConfig" and "browserslist" sections:

```
22 "eslintConfig": {
23   "extends": [
24     "react-app",
25     "react-app/jest"
26   ]
27 },
28 "browserslist": {
29   "production": [
30     ">0.2%",
31     "not dead",
32     "not op_mini all"
33   ],
34   "development": [
35     "last 1 chrome version",
36     "last 1 firefox version",
37     "last 1 safari version"
38   ]
39 },
40 "devDependencies": {
41   "ajv": "^7.2.4",
42   "autoprefixer": "^10.4.20",
43   "postcss": "^8.4.49",
44   "tailwindcss": "^3.4.17"
45 }
46 }
```


Tools, and Techniques



The screenshot displays the Visual Studio Code interface with a project named "project E-2 market". The Explorer sidebar on the left shows the project structure, including a "Back-end" folder containing "package.json". The main editor area shows the content of "package.json" with the following JSON structure:

```
1 {
2   "name": "e-2-market",
3   "version": "1.0.0",
4   "main": "index.js",
5   "type": "module",
6   "scripts": {
7     "dev": "nodemon -r dotenv/config --experimental-json-modules src/ index.js"
8   },
9   "author": "Anurag Nidhi",
10  "license": "ISC",
11  "description": "",
12  "dependencies": {
13    "bcrypt": "^5.1.1",
14    "cloudinary": "^2.5.1",
15    "cookie-parser": "^1.4.7",
16    "cors": "^2.8.5",
17    "dotenv": "^16.4.7",
18    "express": "^4.21.2",
19    "jsonwebtoken": "^9.0.2",
20    "mongoose": "^8.9.5",
21    "mongoose-aggregate-paginate-v2": "^1.1.3",
22    "multer": "^1.4.5-lts.1"
23  },
24  "devDependencies": {
25    "nodemon": "^3.1.9",
26    "prettier": "^3.4.2"
27  }
28 }
```

The status bar at the bottom indicates the current file is "main*", the encoding is "UTF-8", and the language is "JSON".

Tools, and Techniques

3. FRONT-END FRAMEWORK: REACT.js

React.js is a powerful front-end JavaScript framework used for building the user interface. It allows for the development of dynamic, responsive, and interactive web pages, providing an excellent user experience. React's component-based architecture ensures that the code is modular, reusable, and easy to maintain.

4. BACK-END FRAMEWORK: EXPRESS.js

Express.js is a minimalist back-end web framework for Node.js. It simplifies the development of robust APIs, server-side logic, and the handling of HTTP requests. Express is used to build the server architecture and manage routing, middleware, and user authentication.

5. SECURITY: JWT (JSON Web Tokens)

JSON Web Tokens (JWT) is used for implementing secure user authentication. JWT is a compact and self-contained method for securely transmitting information between parties, ensuring that only authenticated users can access certain features of the platform.



Methodology

1. User-Centered Design

Focus on designing an intuitive and easy-to-navigate interface. Conduct surveys and user to the platform meets the needs of both buyers and sellers, enhancing user experience.

2. Authentication and Verification

Implement a secure user registration system, requiring identity verification to ensure legitimacy. This will build trust and reduce the likelihood of fraud and scams.

3. Search and Filter Optimization

Develop an efficient search engine with multiple filters such as category, location, price range, product condition. This will help users quickly find relevant products based on their preferences.

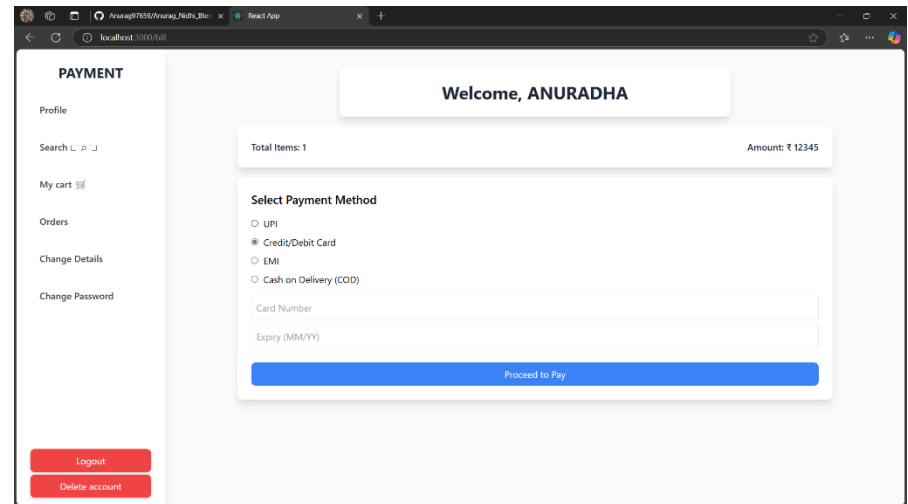
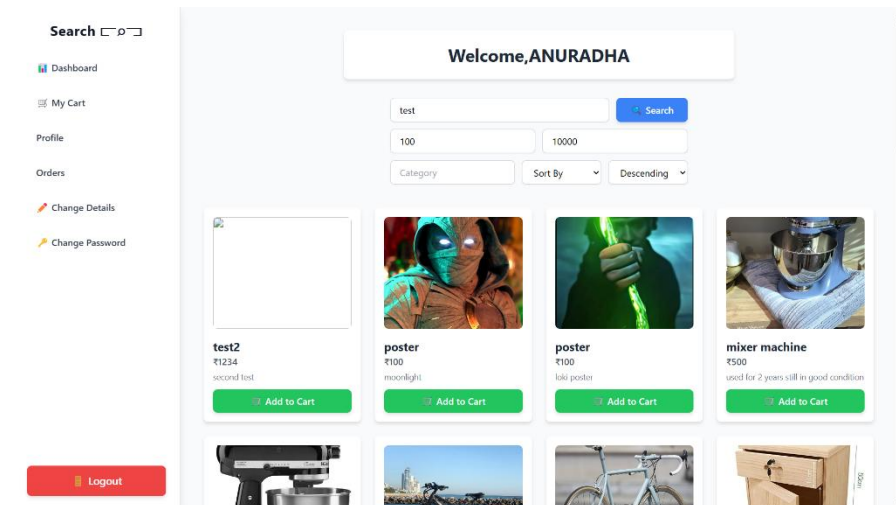
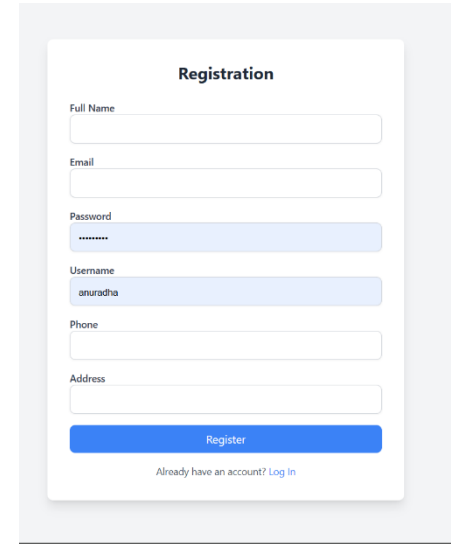
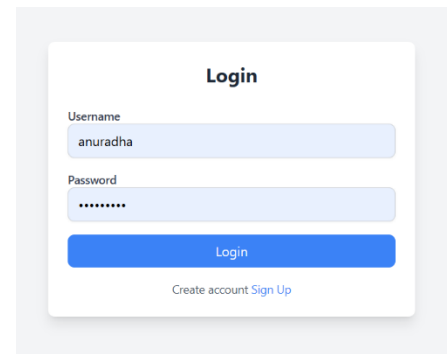
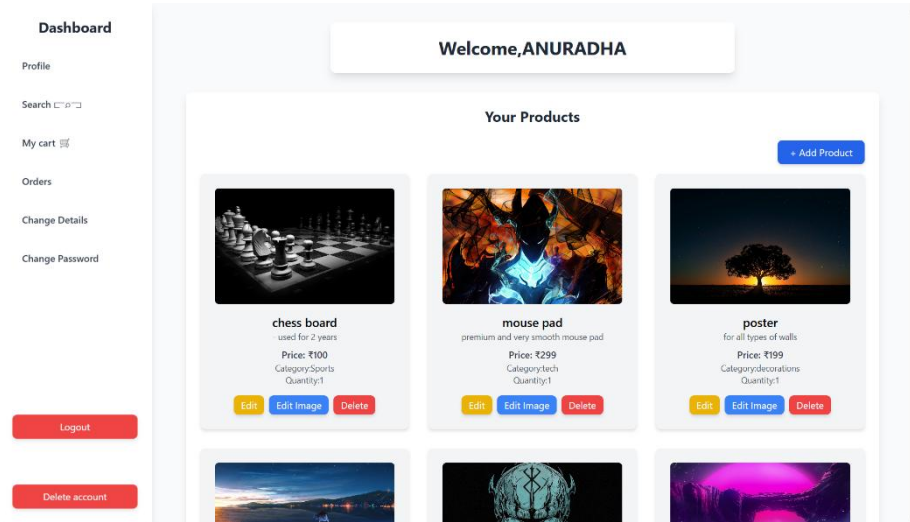
4. Secure Transaction System

Integrate secure payment gateways and offer transaction tracking, ensuring both buyers and sellers can transact safely.

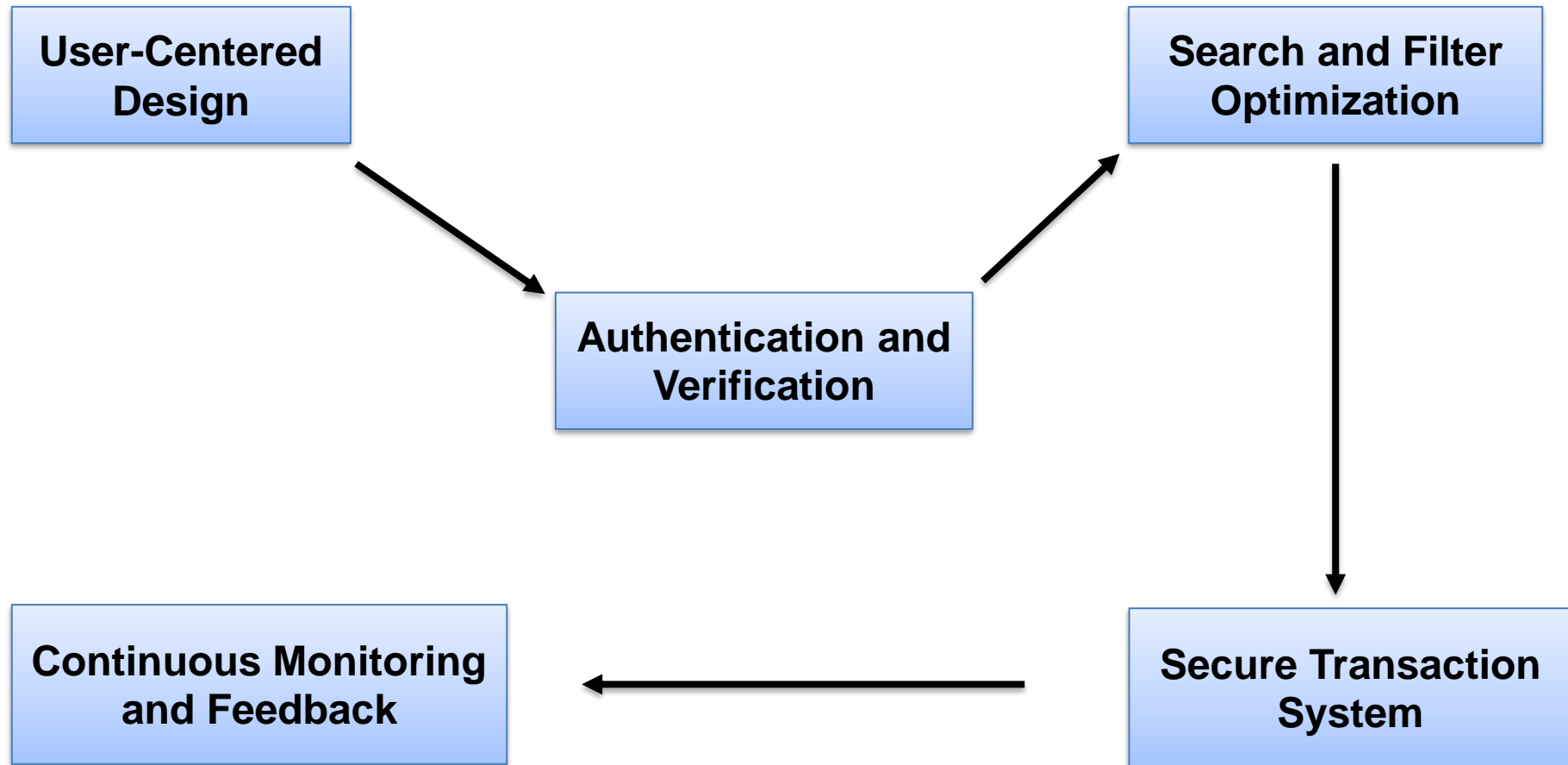
5. Continuous Monitoring and Feedback

Regularly monitor the platform's performance and gather feedback from users to identify areas for improvement. Implement agile development cycles to introduce new features and enhance platform security.

Methodology



Methodology Flowchart



Project Timeline

- 1. Project Planning & Requirements Gathering (1 week):** Define goals, gather requirements, finalize tech stack.
- 2. System Design & Architecture (1 week):** Design database schema, wireframes, and architecture diagrams.
- 3. Frontend Development (2 weeks):** Set up React, develop UI, implement product listing, search, and user profiles.
- 4. Backend Development (1 week):** Set up Node.js server, implement user authentication, APIs for product management, and payment integration.
- 5. Database Setup & Integration (1 week):** Set up MongoDB, integrate collections (users, products, orders).

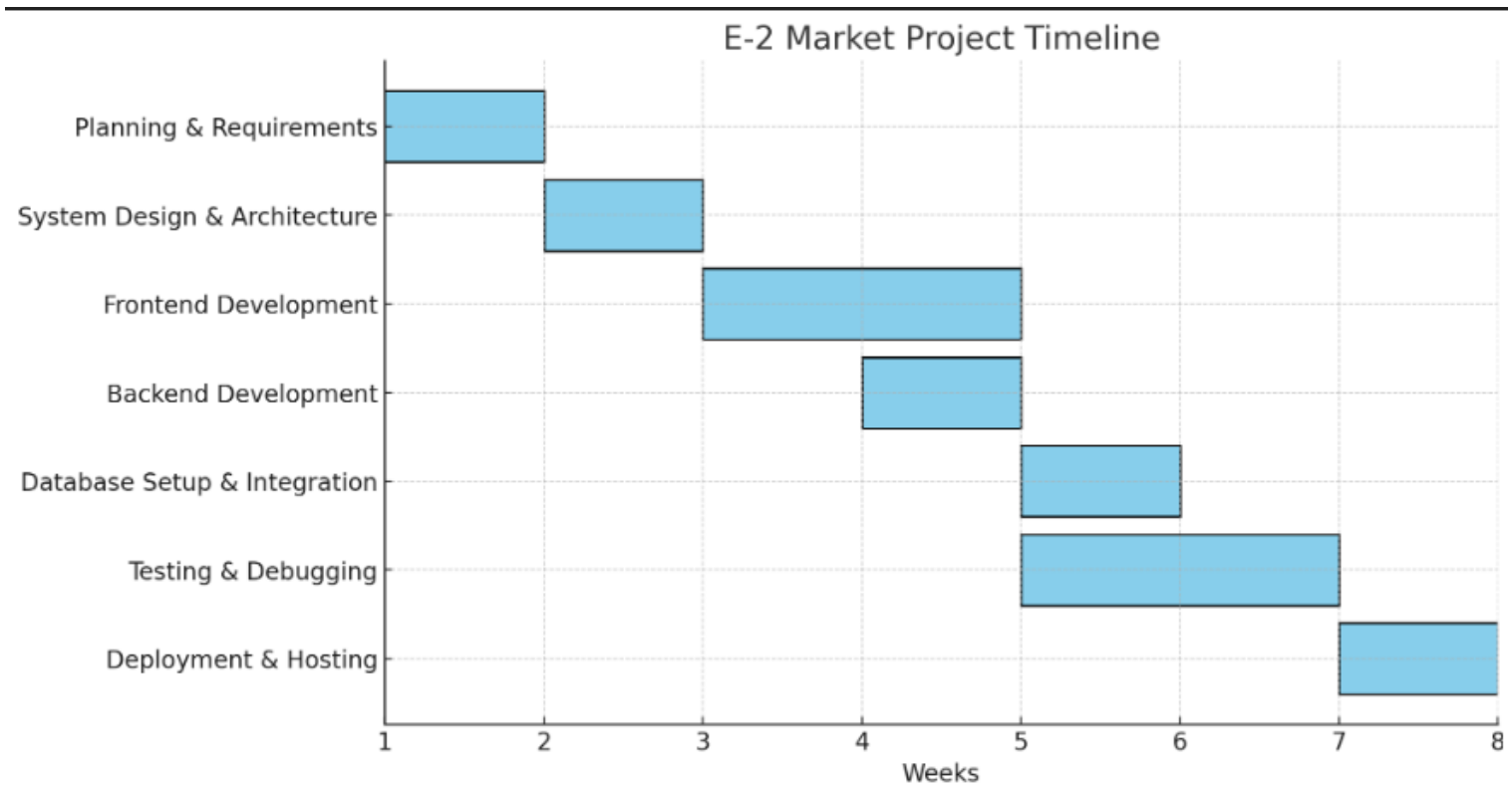


Project Timeline

- 6. Testing & Debugging (2 weeks):** Conduct unit tests, frontend and backend testing, fix bugs.
- 7. Deployment & Hosting (1 week):** Deploy backend and frontend, configure CI/CD and domain.



Project Timeline



Expected Results & Impact

1. Outcomes

- A fully functional online marketplace for buying and selling used products.
- A secure and easy-to-use platform where users can list, browse, and purchase items.

2. Effectiveness

- Solves key problems** like security, trust, and ease of use.
- Key features** include search filters, secure payments, user profiles, and a smooth UI.

3. Deliverable

- A **working prototype** with:
 - Frontend:** React app
 - Backend:** Node.js server, MongoDB database
- A live demo showcasing product listing, search, filtering, and purchases.

4. Impact

For the e-commerce industry: A new model for second-hand product platforms.

For businesses: Helps sell second-hand or refurbished goods.

For users: A cost-effective and reliable way to buy and sell used items.

5. Future Scope

Mobile App: Expand to iOS and Android for better accessibility.



References

Websites:

1. MongoDB Documentation. (2024). *MongoDB, Inc.*
Retrieved from <https://www.mongodb.com/docs/>
2. React Documentation. (2024). *React*. Retrieved from
<https://reactjs.org/docs/getting-started.html>
3. Node.js Documentation. (2024). *Node.js Foundation*.
Retrieved from <https://nodejs.org/en/docs/>



THANK YOU

