**UrbanKicks E-commerce Website Design Document**

**1. Introduction**

UrbanKicks is an E-commerce website designed for selling shoes. The platform provides a user-friendly interface for customers to browse, search, and purchase shoes of various styles and brands. The website is built using the MERN stack (MongoDB, Express.js, React.js, Node.js) and utilizes Tailwind CSS for designing the frontend.

**2. Functional Requirements**

Customer-Facing Features:

1. Browse and search for shoes by category, brand, etc.

2. View detailed product information including images, descriptions, prices, and availability.

3. Add products to the shopping cart and proceed to checkout.

4. Register and login to user accounts for personalized experiences.

5. Track order status and view order history.

6. Update user profile details.

**Admin Features:**

1. Add, edit, and delete products.

2. Manage orders including updating order status and deleting orders.

**3. Non-Functional Requirements**

1. Performance: The website should load quickly and handle concurrent user requests efficiently.

2. Security: Implement authentication and authorization mechanisms to protect user data and prevent unauthorized access.

3. Scalability: Design the system to accommodate potential growth in both users and products.

4. User Experience: Ensure the website is intuitive, responsive, and accessible across different devices and screen sizes.

5. Maintainability: Write clean, modular, and well-documented code to facilitate future updates and maintenance.

**4. Architecture View**

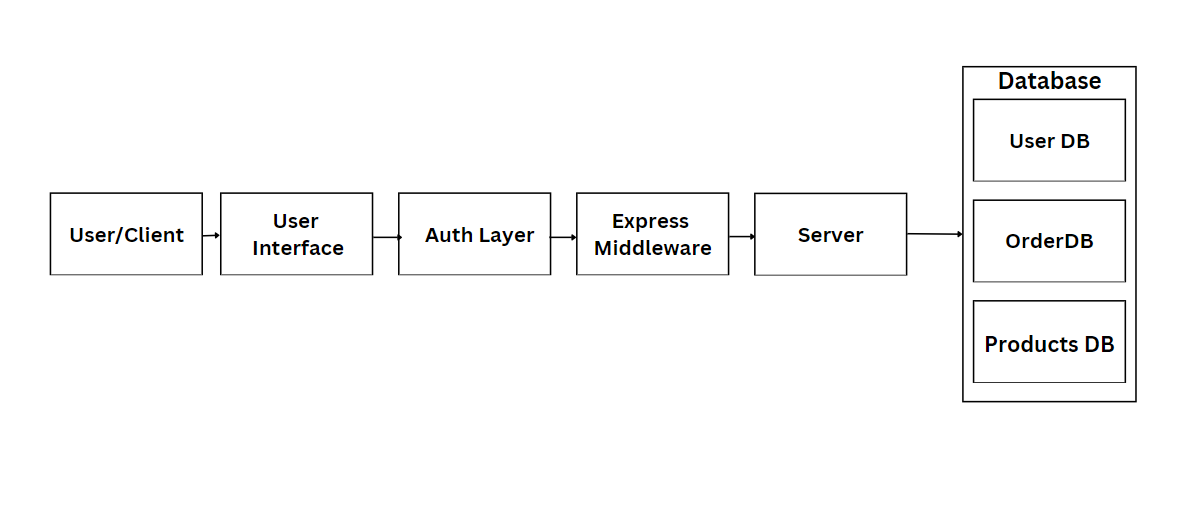
The architecture of the UrbanKicks website follows a typical MERN stack architecture:

- Frontend : React.js for building the user interface.

- Backend : Node.js and Express.js for handling server-side logic and API requests.

- Database : MongoDB for storing product, user, and order data.

- Deployment : The application is deployed on a cloud platform such as AWS, Azure, or Heroku.



**5. Database Design**

The MongoDB database is structured to support the following collections:

1. Products : Stores information about shoes including name, brand, category, price, size, color, and availability.

2. Users : Contains user account details such as username, email, password hash, and role.

3. Orders : Stores order details including products, quantities, total price, and status.

**6. API Design**

The API routes are designed to handle various operations related to products, orders, and users. These routes are secured using authentication middleware for sensitive operations.

**Order Routes**

* + **GET /orders/ -**  Retrieve all orders.
  + **POST /orders/ -** Create a new order.
  + **GET /orders/:id -** Retrieve an order by its ID.
  + **PUT /orders/:id -** Update the status of an order.
  + **DELETE /orders/:id -** Delete an order by its ID.

**Product Routes**

* + **GET /products/ -** Retrieve all products.
  + **GET /products/:id -** Retrieve a product by its ID.
  + **GET /products/category/:category -** Retrieve products by category.
  + **POST /products/ -** Create a new product.
  + **PUT /products/:id -** Update a product by its ID.
  + **DELETE /products/:id -** Delete a product by its ID.

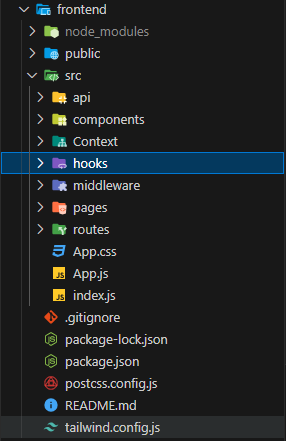
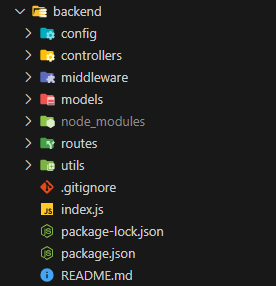
**User Routes**

* + **POST /users/register -** Register a new user.
  + **POST /users/login -** Log in a user.
  + **POST /users/logout -** Log out a user.
  + **GET /users/details** - Retrieve user details.

**7. Frontend Design**

The frontend of UrbanKicks is designed using React.js and styled with Tailwind CSS to create a modern and visually appealing user interface. Components are organized hierarchically to ensure reusability and maintainability.

**Directory structure for Frontend and Backend respectively**

**8. Security**

- Authentication : Implement JWT-based authentication to securely manage user sessions.

- Authorization : Use role-based access control (RBAC) to restrict access to admin functionalities.

- Input Validation : Sanitize and validate user input to prevent injection attacks and other security vulnerabilities.

- Secure Communication : Use HTTPS protocol to encrypt data transmission between client and server.

**9. Deployment**

The UrbanKicks website can be deployed on a cloud platform such as AWS, Azure, or Heroku. Continuous integration and deployment (CI/CD) pipelines can be set up to automate the deployment process.

**10. Maintenance and Support**

Regular maintenance and updates are essential to ensure the website remains functional, secure, and up-to-date with the latest technologies. This includes monitoring for bugs, performance optimizations, security patches, and feature enhancements. Additionally, providing customer support and addressing user feedback are crucial for improving the overall user experience.

**11. Conclusion**

UrbanKicks is a feature-rich E-commerce website designed for selling shoes. By leveraging the MERN stack and Tailwind CSS, the platform offers a seamless shopping experience for customers while providing efficient management tools for administrators. With a focus on performance, security, scalability, and user experience, UrbanKicks aims to become a leading destination for shoe enthusiasts worldwide.