**SHOPEZ: E – COMMERCE**

**APPLICATION**

# **1.Introduction**

* **Project Title :**

ShopeZ : E - Commerce Application

* **Team Members :**

Aarthi R (Frontend)

Aswini S (Backend)

Naveen Kumar S (Database)

Pallavi B (Security)

**2.Project Overview**

* **Purpose :**

Shopez is an innovative e-commerce platform designed to provide a seamless and efficient online shopping experience for users across multiple categories such as fashion, electronics, home goods, beauty products, and more. The platform's primary objective is to offer a user-friendly, secure, and scalable environment where consumers can browse, compare, and purchase products easily, while also providing businesses and vendors with an efficient way to manage their products, sales, and customer interactions.

* **Features :**

1. Convenience: Allow users to shop anytime and anywhere, using desktop and mobile devices, ensuring a fast, responsive, and accessible platform.
2. Variety: Provide a wide selection of products from various categories, ensuring that there’s something for every consumer need and preference.
3. User Engagement: Incorporate features like personalized recommendations, wish lists, and dynamic search filters to enhance customer satisfaction and engagement.
4. Efficiency for Sellers: Equip vendors with a comprehensive set of tools to list and manage products, track inventory, process orders, and gain insights into sales performance.
5. Security: Ensure secure online transactions and personal data protection, adhering to best practices in cybersecurity and privacy.

**3.Architecture :**

**1. Frontend:**

Web application: built using HTML, CSS, JavaScript, and a framework like React or Angular.

**Mobile application:** built using a framework like React Native or Flutter.

Responsive design for a seamless user experience across devices.

**2. Backend:**

API Gateway: handles incoming requests and routes them to appropriate services.

**Services:**

Product Service: manages product information, inventory, and pricing.

User Service: handles user authentication, authorization, and profile management.

Order Service: manages order processing, payment, and fulfillment.

Payment Gateway: integrates with third-party payment providers.

**3. Database:**

Relational database (e.g., MySQL): stores product information, user data, and order history.

NoSQL database (e.g., MongoDB): stores product metadata, user preferences, and session data.

**4. Infrastructure:**

Cloud hosting (e.g., AWS, Google Cloud): provides scalability, reliability, and security.

Load Balancer: distributes traffic across instances for high availability.

Caching: improves performance by reducing database queries.

**5. Security:**

Authentication: implements OAuth, JWT, or session-based authentication.

Authorization: uses role-based access control (RBAC) for user permissions.

Data encryption: protects sensitive data like passwords and credit card numbers.

**6. Third-party integrations:**

Payment gateways (e.g., PayPal, Stripe).

Logistics and shipping providers (e.g., USPS, UPS).

Social media platforms (e.g., Facebook, Instagram).

This architecture provides a solid foundation for building a scalable, secure, and user-friendly e-commerce application like Shopez. However, please note that specific requirements may vary depending on your business needs and technology stack.

**4.Setup Instruction :**

**Technical Prerequisites:**

**1. Programming languages:** Proficiency in JavaScript, HTML, CSS, and a framework like React or Angular.

**2. Backend technology:** Experience with Node.js, Express.js, or a similar backend framework.

**3. Database management:** Knowledge of relational databases (e.g., MySQL) and NoSQL databases (e.g., MongoDB).

**4. Cloud platforms:** Familiarity with cloud hosting services like AWS or Google Cloud.

**5. Security measures**: Understanding of authentication, authorization, and data encryption techniques.

**Non-Technical Prerequisites:**

**1. Business understanding:** Familiarity with e-commerce business models and operations.

**2. Market research:** Knowledge of target audience, market trends, and competitors.

**3. Product management:** Experience with product catalog management, inventory control, and order fulfillment.

**4. Customer support:** Understanding of customer service principles and support channels.

**5. Project management:** Ability to manage timelines, resources, and stakeholders.

**Additional Prerequisites:**

**1. Version control**: Proficiency with Git and GitHub.

**2. Agile development:** Experience with iterative development methodologies like Scrum or Kanban.

**3. Testing and QA:** Knowledge of testing frameworks and quality assurance processes.

1. **4. DevOps:** Familiarity with continuous integration and continuous deployment (CI/CD) pipelines.

**5. Collaboration tools:** Proficiency with communication and collaboration platforms like Slack or Asana.

By meeting these prerequisites, you'll be well-equipped to build and maintain a successful e-commerce application like Shopez. Remember to stay up-to-date with industry trends and continuously improve your skills to ensure the application's growth and scalability.

**Installation**

Here is a step-by-step guide to installing the Shopez e-commerce application:

**1. Clone the repository:** Clone the Shopez repository from GitHub using the command `git clone (link unavailable)

**2. Install dependencies**: Install the required dependencies by running the command npm install or yarn install

**3. Set up environment variables:** Create a .env file in the root directory and add the following variables:

- DB\_HOST: database host

**-** DB\_USER: database username

- DB\_PASSWORD: database password

- DB\_NAME: database name

- PORT: application port

**4. Create database:** Create a new database using the command mysql -u root -p<password> <database\_name>

**5. Run migrations:** Run the migrations using the command npx sequelize-cli db:migrate

**6. Seed database:** Seed the database with initial data using the command npx sequelize-cli db:seed

**7. Start application:** Start the application using the command npm start or yarn start

**8. Access application:** Access the application by visiting http://localhost:3000 in your web browser.

**5.Folder Structure :**

* **Client**

1. public (static assets)
2. images
3. styles
4. scripts
5. Src (client-side code)
6. components (React components)
7. containers (React containers)
8. actions (Redux actions)
9. reducers (Redux reducers)
10. utils (utility functions)
11. api (API integrations)
12. config (client config)
13. index.js (client entry point)

* **Server**

1. routes (API routes)
2. controllers (business logic)
3. models (database models)
4. services (external services integrations)
5. index.js (server entry point)
6. app.js (Express app)
7. Shared
8. database (database schema and migrations)
9. schema
10. migrations
11. tests (unit tests and integration tests)
12. client
13. server
14. package.json (project metadata and dependencies)

This structure separates the client and server code into different folders, making it easier to manage and maintain. The shared folder contains code that is shared between the client and server, such as database schema and tests.

**6. Running the application :**

**Frontend (Client):**

npm start or yarn start

This command will start the development server for the frontend application, which is usually accessible at http://localhost:3000.

Backend (Server):

- node server/index.js or nodemon server/index.js

This command will start the backend server, which is usually accessible at http://localhost:5000.

Alternatively, you can use a single command to start both servers concurrently:

npm run dev or yarn dev

This command will start both the frontend and backend servers simultaneously, using a tool like concurrently.

**7.API Documentation :**

Endpoints

Products

- GET /products

- Retrieve a list of all products

- Parameters: none

- Example Response:

- Status Code: 200

- Data: [

{

d: 1,

ame: "Product 1",

escription: "This is product 1",

rice: 10.99,

mage: "product1.jpg"

},

...

]

- GET /products/:id

- Retrieve a specific product by ID

- Parameters: id (integer)

- Example Response:

- Status Code: 200

- Data: {

id: 1,

name: "Product 1",

description: "This is product 1",

price: 10.99,

image: "product1.jpg"

}

- POST /products

- Create a new product

- Parameters: name (string), description (string), price (number), image (string)

- Example Response:

- Status Code: 201

- Data: {

id: 1,

name: "Product 1",

description: "This is product 1",

price: 10.99,

image: "product1.jpg"

}

- PUT /products/:id

- Update a specific product

- Parameters: id (integer), name (string), description (string), price (number), image (string)

- Example Response:

- Status Code: 200

- Data: {

id: 1,

name: "Product 1",

description: "This is product 1",

price: 10.99,

image: "product1.jpg"

}

- DELETE /products/:id

- Delete a specific product

- Parameters: id (integer)

- Example Response:

- Status Code:

**8. Authentication :**

Authentication and authorization are handled in the Shopez e-commerce application using a combination of tokens and sessions.

**Authorization :**

When a user makes a request to a protected route, the token is sent in the request headers.

The server verifies the token and checks if it is valid and has not expired.

If the token is valid, the user is granted access to the requested resource.

**Tokens :**

Tokens are generated using a secret key and are signed with a hashing algorithm (HS256).

Tokens contain the user's ID and other relevant information.

Tokens are set to expire after a certain amount of time (e.g. 1 hour).

**Sessions :**

Sessions are used to store the user's token and other relevant information.

Sessions are implemented using a session management library (e.g. Express-Session).

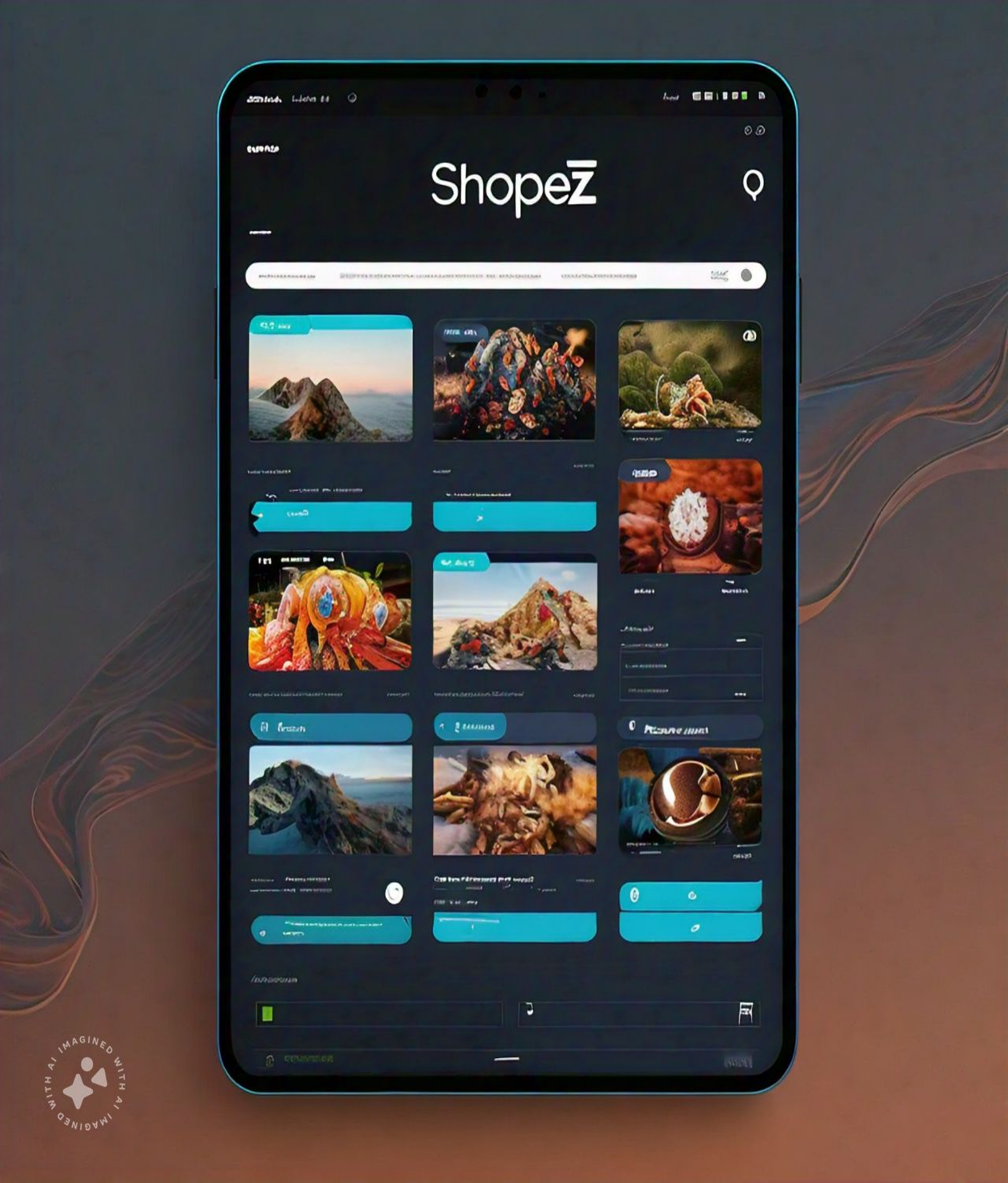
**Other methods :**

Password hashing: Passwords are hashed using a hashing algorithm (BCrypt) before being stored in the database.

CSRF protection: CSRF tokens are generated and verified for each request to prevent cross-site request forgery attacks.

Overall, the Shopez e-commerce application uses a combination of tokens and sessions to handle authentication and authorization. Tokens are used to verify the user's identity and grant access to protected routes, while sessions are used to store the user's token and other relevant information. Additionally, password hashing and CSRF protection are used to ensure the security of user data.

**9. User Interface :**

****

**10. Testing :**

**Unit Testing:** Test individual components and functions to ensure they work as expected.

**Integration Testing:** Test how components interact with each other.

**System Testing:** Test the entire application to ensure it meets requirements.

**Acceptance Testing:** Test the application with real-world scenarios to ensure it meets user expectations.

**Tools Used:**

Unit Testing: Jest, Enzyme

Integration Testing: Jest, Cypress

System Testing: Cypress, Selenium

Acceptance Testing: Cypress, TestRail

Additionally, we will also use:

Continuous Integration: Jenkins, Travis CI

Continuous Deployment: Docker, Kubernetes

Code Review: GitHub, Bitbucket

Code Analysis: SonarQube, CodeCoverage

**Testing Approach:**

**Black Box Testing:** Test the application without knowing the internal workings.

**White Box Testing:** Test the application with knowledge of the internal workings.

**Gray Box Testing:** Test the application with limited knowledge of the internal workings.

**Test Cases:**

User Registration and Login

Product Search and Filtering

Product Details and Reviews

Shopping Cart and Checkout

Payment Gateway Integration

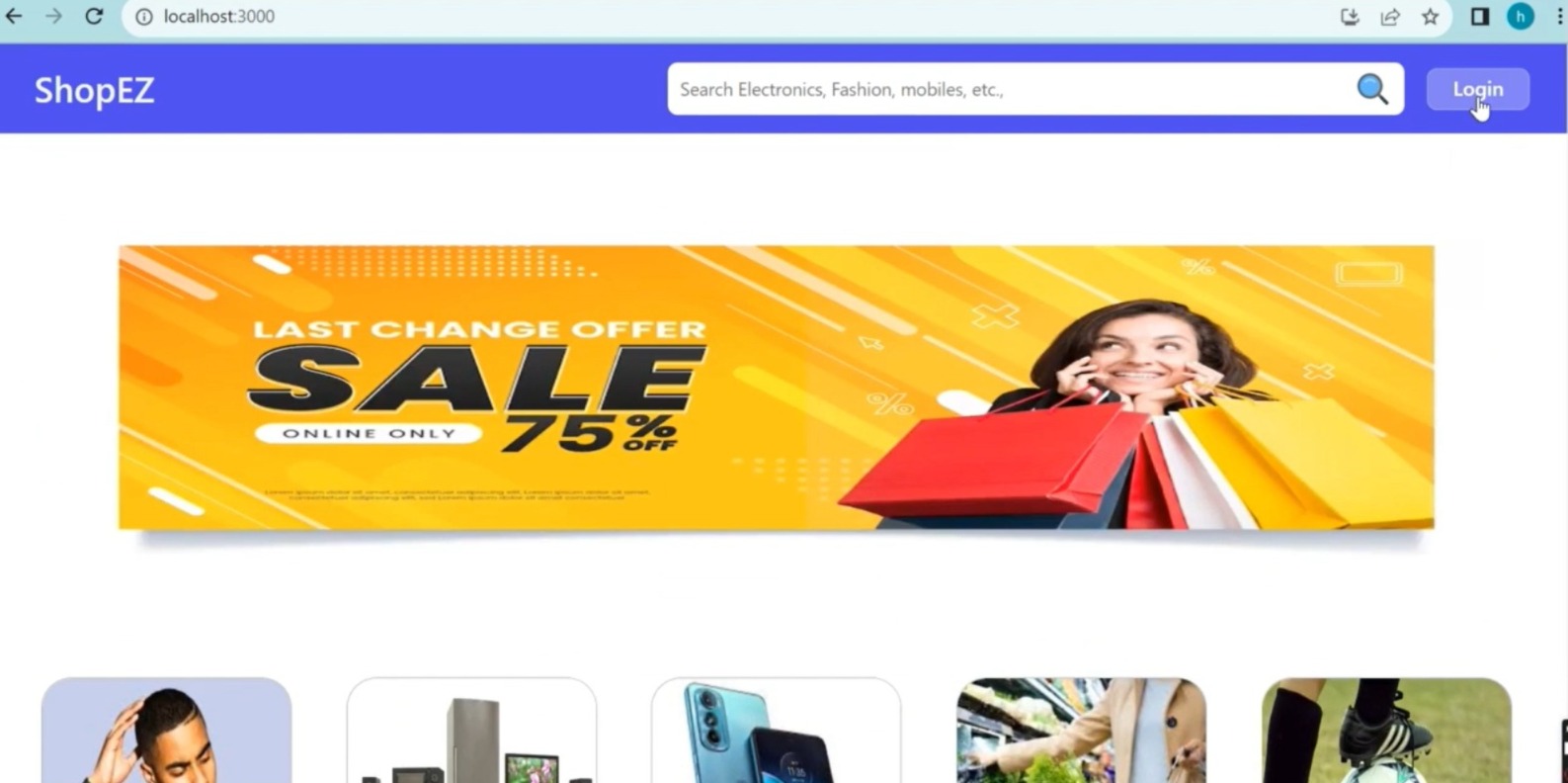
Order Management and Tracking

User Profile and Account Management

Search and Navigation

Mobile Responsiveness and Accessibility

**11. Screenshot or Demo :**



**12. Known Issues :**

**1.Search Functionality:** The search bar does not support special characters, causing errors in search results.

**2. Product Images:** Product images may not display correctly on mobile devices, causing layout issues.

**3. Checkout Process:** Users have reported issues with the checkout process, including errors with payment processing and order confirmation.

**4. Mobile Responsiveness:** Certain pages may not be fully responsive on mobile devices, causing difficulties with navigation.

**5. Browser Compatibility:** The website may not be fully compatible with older browsers, causing display issues and errors.

**6. Product Reviews:** Users have reported issues with posting product reviews, including errors with submitting and displaying reviews.

**7. Shopping Cart:** Users have reported issues with updating and removing items from the shopping cart.

**8. Payment Gateway:** Users have reported issues with payment processing, including errors with credit card transactions.

**9. Order Tracking:** Users have reported issues with tracking orders, including errors with order status updates.

**10. Website Speed:** Users have reported issues with website speed, including slow loading times and delays.

**13. Future Enhancement :**

**1. Personalized Recommendations:** Implement AI-powered personalized product recommendations based on user behavior and preferences.

**2. Enhanced Mobile App:** Develop a mobile app for ShopEZ to provide a seamless shopping experience on-the-go.

**3. Social Media Integration:** Integrate social media platforms to allow users to share products and promotions.

**4. Advanced Analytics:** Implement advanced analytics to provide insights into user behavior and sales trends.

**5. Improved Search Functionality:** Enhance search functionality with features like autocomplete, spell-check, and faceted search.

**6. Product Videos:** Allow sellers to upload product videos to showcase products from different angles.

**7. Customer Reviews and Ratings:** Implement a review and rating system for products and sellers.

**8. Loyalty Program:** Develop a loyalty program to reward repeat customers.

**9. Multi-Language Support:** Add support for multiple languages to cater to a global customer base.

**1. AR/VR Integration:** Integrate Augmented Reality (AR) and Virtual Reality (VR) features to enhance product visualization.

**2. Subscription-Based Services:** Offer subscription-based services for regular deliveries of essentials.

**3. Partnerships and Collaborations:** Collaborate with influencers, brands, and other e-commerce platforms to expand offerings.

**4. Enhanced Security:** Implement advanced security measures to protect user data and prevent cyber threats.

**5. Simplified Checkout Process:** Streamline the checkout process with features like guest checkout and one-click payments.

**6. Product Customization:** Allow users to customize products with various options and upgrades.

These potential future features and improvements can enhance the user experience, increase customer satisfaction, and drive business growth for ShopEZ.