Technical Plan for Nike Shoes E-commerce Website

1. Objective and Scope

Objective: To develop a fully functional, user-friendly, and responsive e-commerce website for Nike shoes, enabling customers to browse, search, and purchase products seamlessly.

Scope:

- Frontend design and development.
- Backend system for managing products, users, and orders.
- Payment gateway integration.
- User authentication and authorization.
- Responsive design for desktop, tablet, and mobile devices.
- Deployment and maintenance.

---

2. Requirements

Functional Requirements:

- **User Features:**
  - Account registration and login.
  - Product browsing and searching.
  - Add to cart, checkout, and order tracking.
  - Wishlist functionality.
  - Product reviews and ratings.
- **Admin Features:**
  - Manage products (add, update, delete).
  - Monitor orders and customer data.
  - Generate sales reports.

Non-Functional Requirements:

- High performance and fast page load times.
- Secure transactions and data handling.
- Scalable architecture to handle high traffic.
- Accessibility compliance.

---

3. Architecture and Design

High-Level Architecture:

- **Frontend:** Next.js framework with TypeScript and Tailwind CSS.
- **Backend:** Node.js with Express.js.
- **Database:** PostgreSQL for relational data and Redis for caching.
- **Payment Gateway:** Stripe for secure payments.
- **Hosting and Deployment:** Vercel for frontend and AWS for backend services.

Design Considerations:

- Modular and reusable components for frontend UI.
- RESTful APIs for communication between frontend and backend.
- SEO-friendly structure for better search engine visibility.

---

4. Technologies and Tools

- **Frontend:**
  - Next.js, React, TypeScript, Tailwind CSS.
  - Figma for UI/UX design.
- **Backend:**
  - Node.js, Express.js.
  - Stripe API for payments.
- **Database and Storage:**
  - PostgreSQL for data storage.
  - AWS S3 for media storage.
- **Version Control:** Git and GitHub.
- **Testing:** Jest for unit testing and Cypress for end-to-end testing.
- **Project Management:** Jira or Trello.

---

5. Resource Planning

Team Roles:

- **Project Manager:** Oversees project progress and coordination.
- **UI/UX Designer:** Designs user interface and experience.
- **Frontend Developer:** Builds the client-side application.
- **Backend Developer:** Develops server-side logic and APIs.
- **Database Administrator:** Manages the database.
- **Quality Assurance (QA):** Ensures product quality.

Budget Allocation:

- Hosting and deployment services.
- Licenses for tools and services (e.g., Stripe, AWS).
- Salaries for the development team.

---

6. Timeline and Milestones

Phase	Duration	Deliverables
Planning and Design	2 weeks	Wireframes, technical plan.
Frontend Development	4 weeks	Functional UI components.
Backend Development	4 weeks	API endpoints and database setup.
Testing and QA	2 weeks	Bug-free application.
Deployment	1 week	Live e-commerce website.

---

7. Risk Assessment

Potential Risks:

- High traffic may lead to server crashes.
- Security vulnerabilities in payment processing.
- Delays in integrating third-party APIs.

Mitigation Strategies:

- Use load balancers and caching mechanisms.
- Implement SSL/TLS for secure connections.
- Perform regular code reviews and security audits.

---

8. Testing and Quality Assurance

- **Unit Testing:** Test individual components and functions.
- **Integration Testing:** Validate interactions between different modules.
- **Performance Testing:** Ensure website can handle high traffic.
- **Security Testing:** Validate secure data handling.
- **User Acceptance Testing (UAT):** Ensure the site meets user expectations.
