



# HACKATHON

## HACKATHON: MILESTONE-BASED E-COMMERCE APP WITH NEXT.JS

Event Date: 6th September 2024  
Duration: 24 Hours (Starting Midnight after the event)  
Submission Deadline: 7th September 2024, Midnight

Ameen Alam  
Dean of Faculty

**Overview:**

This hackathon challenges you to build a complete E-Commerce Application using **Next.js**, divided into five key milestones. Each milestone builds upon the previous one, starting from setting up the store to implementing dynamic features such as authentication and payment integration. You can work from home, completing as many milestones as you can within the 24-hour time frame.

---

**Hackathon Structure:**

- The hackathon is divided into 5 milestones.
  - Each milestone carries 100 points.
  - You can submit your progress for each completed milestone.
  - Aim to complete as many milestones as possible within the 24-hour limit.
- 

**Milestone 1: Static E-Commerce Homepage using Next.js (100 Points)****Objective:**

Create a static homepage using **Next.js** to display product listings, categories, and navigation.

**Requirements:**

- Build a homepage using **Next.js Pages** and **Static Generation**.
- Fetch product data from a mock API or static file using **getStaticProps()**.
- Design sections like:
  - Product Listings
  - Categories
  - Navigation Bar
  - Footer
- Ensure the page is responsive and SEO-friendly.

**Scoring Breakdown:**

- **Next.js Page Structure and Static Generation (40 points):**  
Is the homepage built using the correct Next.js functions like `getStaticProps`?
- **CSS Styling and Responsiveness (40 points):**  
Does the homepage look visually appealing and function across mobile and desktop devices?

- **Basic Interactivity (20 points):**  
Is there interactivity, such as a button to filter product categories?
- 

## **Milestone 2: Add-to-Cart Functionality with State Management (100 Points)**

### **Objective:**

Implement a cart functionality using Next.js and manage the cart state across pages.

### **Requirements:**

- Use **React Context API** or **Redux** for managing the cart state.
- Users should be able to add/remove products from the cart.
- Ensure cart persistence across pages using Next.js routing and **LocalStorage**.

### **Scoring Breakdown:**

- **Cart State Management (50 points):**  
Is the cart state properly managed and persistent across different pages?
  - **Next.js Routing (30 points):**  
Are you using **Next.js Link** for navigation between the homepage, product pages, and cart?
  - **Cart UI and Functionality (20 points):**  
Is the cart UI user-friendly and does it update in real-time?
- 

## **Milestone 3: Dynamic Product Detail Pages using Next.js API Routes (100 Points)**

### **Objective:**

Create dynamic product detail pages with user reviews using **Next.js Dynamic Routing** and **API Routes**.

### **Requirements:**

- Implement dynamic routing with **[productId].js** for product detail pages.
- Fetch product data using **getStaticPaths** and **getStaticProps**.
- Implement reviews functionality using **Next.js API routes** for submitting and displaying reviews.
- Ensure that users can add reviews without refreshing the page using **Client-Side Data Fetching**.

### **Scoring Breakdown:**

- **Dynamic Routing and Static Generation (40 points):**  
Is the product detail page built using Next.js dynamic routing and pre-rendering methods?

- **Reviews System (40 points):**  
Can users successfully submit and view reviews in real-time?
  - **User Experience (20 points):**  
Is the product page user-friendly and interactive?
- 

#### **Milestone 4: User Authentication using NextAuth.js (100 Points)**

##### **Objective:**

Integrate authentication for users using **NextAuth.js** to allow for login and registration.

##### **Requirements:**

- Implement **NextAuth.js** for secure user authentication.
- Allow users to register, log in, and access their account.
- Ensure authenticated users can proceed to the checkout.
- Form validation for login/registration.

##### **Scoring Breakdown:**

- **Authentication Flow (60 points):**  
Can users register, log in, and access their account securely with proper validation?
  - **Next.js Integration with NextAuth.js (40 points):**  
Is NextAuth.js implemented correctly, and are protected routes handled properly?
- 

#### **Milestone 5: Payment Integration and Order Management (100 Points)**

##### **Objective:**

Integrate a payment gateway using a mock API and manage the order process.

##### **Requirements:**

- Set up payment processing using a mock payment API.
- Use **Next.js API Routes** to handle backend order and payment processing.
- Show an order confirmation page after successful payment, displaying the order details.
- Generate an invoice for the user and allow them to download it.

##### **Scoring Breakdown:**

- **Payment Flow and API Integration (50 points):**  
Does the payment system work properly with a mock API, and are orders processed securely?

- **Order Confirmation and Invoice Generation (50 points):**  
Does the order confirmation page show all necessary details, and can users download their invoice?
- 

#### **Hackathon Timeline:**

**Start Time:** Midnight 06 Sep 2024

**End Time:** Midnight 06 Sep 2024

**Submission Deadline:** Midnight 6 Sep 2024

---

#### **Submission Guidelines:**

- **Milestone 1:** Submit your Next.js code for the static homepage.
  - **Milestone 2:** Submit the code for the cart functionality and state management.
  - **Milestone 3:** Share the live dynamic product detail page link and repository.
  - **Milestone 4:** Submit the authentication system with NextAuth.js and form validation.
  - **Milestone 5:** Submit the project with payment integration, order management, and invoice generation.
- 

#### **Awards and Recognition:**

Students who complete all milestones will be awarded the title of "Senior Developer." Those who finish with high scores will receive special recognition.

---

#### **Evaluation Criteria:**

- **Completeness:** Did you meet the objectives of each milestone?
  - **Code Quality:** Is your code clean, well-organized, and efficient?
  - **Functionality:** Does your e-commerce app work as expected with all features functional?
  - **Design and User Experience:** Is the app visually appealing, responsive, and user-friendly?
- 

**Good luck, and we look forward to seeing your innovative Next.js solutions!**