**1️⃣ Step 1: Choose Running Category**

📌 **User Story:**  
**Title:** Allow runners to select their category (5KM, 10KM, 21KM, etc.)  
**Description:**  
As a **runner**, I want to **select my preferred running category** so that I can proceed with the registration process.

**Front-End Tasks**

✅ Display available categories fetched from the backend  
✅ Highlight selected category and proceed button  
✅ Show category-specific price and rules  
✅ Validate category selection before proceeding

**Back-End Tasks**

✅ Create API to fetch event categories (GET /events/:id/categories)  
✅ Store user-selected category in session or database (POST /registration/category)  
✅ Ensure selected category is valid and available

**2️⃣ Step 2: Enter Contact Information**

📌 **User Story:**  
**Title:** Collect Runner's Contact Information  
**Description:**  
As a **runner**, I want to **provide my contact details** (mobile number, email) so that I can receive event updates and confirmations.

**Front-End Tasks**

✅ Create a form for **mobile number & email** input  
✅ Validate email and phone number formats  
✅ Add OTP verification for mobile number  
✅ Store entered details temporarily

**Back-End Tasks**

✅ API to store contact details (POST /registration/contact-info)  
✅ Implement **OTP verification** (POST /send-otp and POST /verify-otp)  
✅ Store verified mobile number in the database

**3️⃣ Step 3: Enter Personal Details**

📌 **User Story:**  
**Title:** Collect Personal Information of Runners  
**Description:**  
As a **runner**, I want to **enter my personal details** (Name, Age, Gender, Emergency Contact, T-shirt size) so that my registration can be complete.

**Front-End Tasks**

✅ Create input fields for **Name, DOB, Gender, Emergency Contact, T-shirt Size**  
✅ Fetch dynamic T-shirt size options  
✅ Validate age restrictions based on category rules  
✅ Store input temporarily before submission

**Back-End Tasks**

✅ API to store user personal details (POST /registration/personal-info)  
✅ Validate age category mapping (middleware/checkAgeCategory)  
✅ Store data in PostgreSQL database

**4️⃣ Step 4: Payment Process**

📌 **User Story:**  
**Title:** Enable Secure Payment for Registration  
**Description:**  
As a **runner**, I want to **make a secure payment** for my event registration so that my spot is confirmed.

**Front-End Tasks**

✅ Display **order summary** with ticket price & applicable discount  
✅ Integrate **Razorpay payment gateway**  
✅ Show **payment success/failure** messages  
✅ Redirect to success/failure page

**Back-End Tasks**

✅ Generate **order ID** (POST /payment/initiate)  
✅ Integrate **Razorpay API** (POST /payment/callback)  
✅ Store payment status in database  
✅ Handle **refunds & failures**

**5️⃣ Step 5: Success Screen**

📌 **User Story:**  
**Title:** Show Confirmation & Next Steps  
**Description:**  
As a **runner**, I want to **see a confirmation screen** with my details so that I can be assured of my successful registration.

**Front-End Tasks**

✅ Display **Registration ID, Payment Status, Event Details**  
✅ Show **Download BIB & Receipt** option  
✅ Provide **social media share option**  
✅ Allow user to edit minor details if required

**Back-End Tasks**

✅ API to fetch registration details (GET /registration/:id)  
✅ Store registration confirmation email in queue for sending  
✅ Log registration for analytics