

# Dream Team - Task 1

## Chapter 0: Overview

This project is all about getting you up to speed with the awesome technologies we use to develop web applications: **Next.js** for our dynamic and interactive front end and **Django** for our powerful backend.

You'll dive into Next.js's dynamic routes, server-side rendering, image optimization, and server actions. On the Django side, you'll work with models and ORM, and get to grips with designing APIs using GraphQL and the Strawberry package. Plus, we'll touch on how everything fits together in a client-server setup and sprinkle in some DevOps practices to round out your skills.

## Chapter 1: Setting the Stage

Welcome Aboard! Imagine you're entering a world where you're about to build a dynamic online store from scratch. Your goal is to create a platform where users can browse products, add them to their cart, and make purchases, all while ensuring a smooth, secure, and interactive experience.

### Tools of the Trade:

Git: Version control system for managing project files.

GitHub: For hosting the git repository.

Backend: Django with [Chowkidar](#) for authentication.

Frontend: Next.js, using either app-router or pages-router syntax.

API: [Strawberry](#) GraphQL.

Testing:

Backend: Perform unit testing using the [unittest](#) framework.

Frontend: Perform unit testing using the [Vitest](#) framework.

System: Perform end-to-end testing using the [Cypress](#) framework.

## Chapter 2: User Management and Authentication

### Step 1: Registration and Login

Implement user registration and login using the [Chowkidar package](#) in Django. Users will sign up using their email, password, and name. Assign a unique username at the time of creation of a user.

### Step 2: Creating a Profile Page

Implement the user profile models and mutation, which enables users to update their information, including uploading an avatar.

Build a profile page where users can update their information, including uploading an avatar and editing their username.

## **Chapter 3: Elevating Admin Capabilities**

### **Step 3: Admin User Creation**

Interactive CLI Command - Create a Django management command that allows you to create an admin user via the command line. The command should be interactive, asking for the name, email, and password.

### **Step 4: Admin Panel**

Admin Access: Once logged in as an admin, the user should be redirected to an admin panel. This UI should allow the admin to manage the store efficiently.

Product Management: In the admin panel, implement the ability to create, edit, and delete products. These products will be visible to users on the store's public page.

### **Step 5: Order Management**

Order Details: The admin should be able to view the details of every order placed on the platform, including the quantity and items ordered.

## **Chapter 4: Building the User Experience**

### **Step 6: Public Landing Page**

Storefront: Create a landing page for the store that displays the products added by the admin. When a non-logged-in user clicks "Buy Now," a login page should pop up, prompting them to log in before proceeding.

### **Step 7: Product Pages**

Dynamic Routing: Use Next.js dynamic routes to create individual pages for each product. These pages will display product details and provide options to add the item to the cart.

### **Step 8: Shopping Cart**

Implement a shopping cart that syncs across all pages using the React Context API. The cart should update in real-time as users add or remove items.

### **Step 9: Checkout and Order Confirmation Email**

Design a checkout page where users can review their cart and confirm their order. Once confirmed, trigger an email from the backend to the user's email with the order summary.

## **Chapter 5: Deployment**

Now that your platform is complete, try deploying your project using any tools or services you prefer. Choose a deployment method that works best for you and make your platform live!

## **Chapter 6: Platform Enhancement**

Once the above tasks are all completed, if you feel like going the extra mile and adding additional features that you think will enhance the platform, you are highly encouraged to do so.

### **Plagiarism Policies**

It is highly encouraged to use Google search engines and AI solutions like ChatGPT and Gemini to complete these tasks. However, you must be able to explain your code.

### **Deadline**

March 13th 2025