

DAY : 2

TECHNICAL FOUNDATION OF MARKET PLACE

Overview

On Day 2, we are establishing the technical foundation of the marketplace project. This includes setting up essential frameworks, libraries, and tools to ensure smooth development and scalability. The focus is on creating a robust environment for both frontend and backend operations.

Objectives

Integrate backend frameworks and APIs for seamless data handling.

Configure environment variables for secure data management.

Establish database connectivity and define basic schemas.

Set up reusable components and utilities for frontend development.

Progress

1. Backend Setup

Installed [Express.js/Next.js API routes] for handling server-side logic.

Connected the project to [MongoDB/Firebase/PostgreSQL] for database operations.

Created basic schemas/models for:

Users

Products

Orders

Defined a RESTful or GraphQL API structure for CRUD operations.

2. Environment Variables

Configured .env file for sensitive data (e.g., API keys, database URLs):

DATABASE_URL=<your_database_url>

API_KEY=<your_api_key>

3. Frontend Setup

Created reusable components:

Header/Navbar

Footer

Product Card

Installed essential libraries:

Axios/Fetch for API calls

Tailwind CSS/Chakra UI for styling

4. Routing and Navigation

Implemented basic navigation using [Next.js Router/React Router].

Created placeholder routes for:

/ (Home)

/products (Product Listings)

/cart (Shopping Cart)

/checkout (Checkout)

Tech Stack Enhancements

Backend:

Framework: [Express.js/Next.js]

Database: [MongoDB/Firebase/PostgreSQL]

ORM/ODM: [Mongoose/Prisma/Sequelize]

Frontend:

Framework: [React.js/Next.js]

Styling: [Tailwind CSS/Chakra UI]

Installation

To test the updated technical foundation locally:

1. Clone the repository:

`git clone`

2. Navigate to the project directory:

`cd marketplace-project`

3. Install backend and frontend dependencies:

`npm install`

4. Run the backend server:

`npm run server`

5. Start the frontend:

`npm run dev`

Notes

Environment variables are critical. Do not hardcode sensitive information.

Ensure API routes are functioning correctly by testing with tools like Postman or Thunder Client.

Contributing

Contributions are welcome. Fork the repository and submit a pull request for any enhancements or fixes.