

# Project Development Phase

Date	21 February 2025
Team ID	LTVIP2026TMIDS86089
Project Name	Order On The Go: Your On-Demand Food Ordering Solution
Maximum Marks	

## 1. Introduction

- Briefly describe the purpose of the development phase.
- State the tech stack (MERN: MongoDB, Express.js, React.js, Node.js).
- Mention your development environment (VS Code, GitHub, MongoDB Atlas, etc.).

## 2. Development Setup

- **Tools & Frameworks:** List IDE, version control (Git), libraries used (React Router, Mongoose, etc.).
- **Configuration:** Document npm scripts, environment variables, and backend server setup.
- **Dependencies:** Provide a table of major dependencies (e.g., React, Express, Mongoose, Stripe API).

## 3. Module Development

Break down features into modules and explain how you developed them:

- **User Authentication:** Registration, login, JWT tokens, bcrypt for password hashing.
- **Product Catalog:** Fetching products from MongoDB, displaying with React components.
- **Cart Management:** Add/update/remove items, linked to user ID.
- **Order Placement:** Checkout flow, payment gateway integration.
- **Notifications:** Firebase Cloud Messaging for order updates.
- **Admin Dashboard:** CRUD operations for restaurants/products, order monitoring.

```

{
  "name": "server",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "start": "node index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bcrypt": "^5.1.0",
    "body-parser": "^1.0.0",
    "cors": "^2.8.5",
    "dotenv": "^17.3.1",
    "express": "^4.22.1",
    "mongoose": "^7.8.9",
    "parser": "^0.1.4"
  }
}

```

## 4. Testing

- Unit Testing:** API endpoints tested with Postman or Jest.
- Integration Testing:** End-to-end order flow tested.
- Bug Fixes:** Document major issues (e.g., npm script errors, backend crashes) and how you solved them.

```

PS E:\IV-II\Smart Intern\Food-Ordering-App-MERN\client> npm start
> react-scripts start
> react-scripts start

(node:11312) [DEP0176] DeprecationWarning: fs.F_OK is deprecated, use fs.constants.F_OK instead
(Use node --trace-deprecation ... to show where the warning was created)
(node:11312) [DEP0005] DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(node:11312) [DEP0005] DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled with warnings.

[eslint]
src\Components\Login.jsx
Line 2:29:  useState is defined but never used  no-unused-vars

src\Components\Register.jsx
Line 1:29:  useState is defined but never used  no-unused-vars

src\Pages\Customer\Cart.jsx
Line 21:6:  React Hook useEffect has missing dependencies: 'calculateTotalPrice', 'cart', and 'fetchCart'. Either include them or remove the dependency array  react-hooks/exhaustive-deps
Line 50:6:  React Hook useEffect has a missing dependency: 'calculateTotalPrice'. Either include it or remove the dependency array  react-hooks/exhaustive-deps

src\Pages\Customer\CategoryProducts.jsx
Line 3:8:  'Restaurants' is defined but never used  no-unused-vars
Line 18:18:  React Hook useEffect has a missing dependency: 'fetchRestaurants'. Either include it or remove the dependency array  react-hooks/exhaustive-deps
Line 37:13:  Headings must have content and the content must be accessible by a screen reader  jsx-attr/heading-has-content

src\Pages\Customer\IndividualRestaurant.jsx
Line 16:9:  'navigate' is assigned a value but never used  no-unused-vars

Line 31:5:  React Hook useEffect has missing dependency: 'fetchRestaurants'. Either include it or remove the dependency array  react-hooks/exhaustive-deps
Line 61:10:  'sortFilter' is assigned a value but never used  no-unused-vars

```

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows the project structure for "FOOD-ORDERING-APP-MERN". It includes a "client" folder containing "node\_modules", "public", and "src" with files like "index.js", "package-lock.json", "package.json", and "Schema.js". It also includes "README.md".
- PACKAGE.JSON**: The current file is "server/package.json". The code content is as follows:

```
1  {
2   "name": "server",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "type": "module",
7   "scripts": {
8     "start": "node index.js"
9   },
10  ,
11  "keywords": [],
12  "author": "",
13  "license": "ISC",
14  "dependencies": {
15    "bcrypt": "^5.1.0",
16    "body": "^5.1.0",
17    "body-parser": "^1.20.2",
18    "cors": "^2.8.5",
19    "dotenv": "^17.3.1",
20    "express": "^4.22.1",
21    "mongoose": "^7.8.9",
22    "parser": "^0.1.4"
23  }
24}
25
```

- TERMINAL**: Shows the command "npm start" being run from the directory "E:\W-II\Start\Internz\Food-Ordering-App-MERN\server". The output indicates Node.js version 18.0.0 and shows deprecation warnings related to 'punycode' and 'url.parse'.

```
PS E:\W-II\Start\Internz\Food-Ordering-App-MERN\server> npm start
> server@1.0.0 start
> node index.js

(node:4348) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(node:4348) [DEP0040] DeprecationWarning: 'url.parse()' behavior is not standardized and prone to errors that have security implications. Use the WHATWG URL API instead. CVEs are not issued for 'url.parse()' vulnerabilities.
running @ 6001
```

## 5. Deployment

- Local Deployment:** Running on localhost with Node.js.
- Cloud Deployment:** Hosting backend on AWS EC2 / Render, frontend on Netlify/Vercel, database on MongoDB Atlas.
- CI/CD (optional):** GitHub Actions or manual deployment steps.

## 6. Screenshots & Evidence

The screenshot shows the homepage of the SB Foods application. At the top, there is a navigation bar with a search bar containing the placeholder "Search Restaurants, cuisine, etc." and a "Login" button. Below the header, there is a section titled "Popular Restaurants" which is currently empty. A sidebar on the left contains a list of categories: Breakfast, Biriyani, Pizza, Noodles, and Burger, each accompanied by a small thumbnail image. At the bottom of the sidebar, there is a link labeled "All restaurants".

The screenshot shows a page listing items from the SB Foods application. On the left, there is a sidebar with "Filters" and "Sort By" options. The "Sort By" section includes radio buttons for Popularity, low-price, high-price, Discount, and Rating. The "Food Type" section includes checkboxes for Veg, Non Veg, and Beverages, with "Beverages" checked. The "Categories" section includes a checkbox for MilkShakes, which is also checked. The main content area is titled "All Items" and displays two milkshake options: "Vanilla Milkshake" and "Strawberry Milkshake". Each item has a thumbnail image, a name, a brief description, a price, and an "Add item" button.

React App

localhost:3000/restaurant/69a17b5bd345e54ec14f1d74

## SB Foods

Search Restaurants, cuisine, etc.

Login

### All Items

**Filters**

Sort By

- Popularity
- low-price
- high-price
- Discount
- Rating

Food Type

- Veg
- Non Veg
- Beverages

Categories

- MilkShakes



**Biryani**  
Chicken Biryani with best...  
₹ 200 200  
[Add item](#)



**Chicken 65**  
Deep fried marinated chic...  
₹ 249 249  
[Add item](#)



**Fish-vanjarim**  
Fish-with good quality...  
₹ 250 250  
[Add item](#)