Online restaurant - webapplication that seamlessly integrates Angular and Django to bring you a delightful dining experience. Whether you're a customer exploring the diverse menu or a manager overseeing your restaurant's offerings, our platform is designed to cater to your every need. Project was deployed as docker-compose containers which allows to run webapp from one place with predefined settings.

# Features

**For Customers**

**Explore the Menu:** Browse through a variety of menu categories and discover a rich selection of dishes, each with its own unique flavor profile.

**User-Friendly Cart:** Effortlessly add your favorite dishes to the cart, manage your selections, and proceed to a seamless checkout experience.

**Order with Ease:** Place your order confidently, and track your order history for a convenient and enjoyable dining experience.

**For Managers**

**Efficient Menu Management:** Easily add, edit, or delete dishes, manage prices, and update dish descriptions to keep your menu fresh and enticing.

**Insightful Order Tracking:** Gain valuable insights into customer orders, ensuring efficient management of your restaurant's operations.

**How to Get Started**

**Sign Up:**

Customers can create accounts to manage their orders, while managers have exclusive access to additional features.

**Explore the Menu:**

Navigate through our intuitive menu interface to discover culinary delights tailored to your tastes.

**Place an Order:**

Customers can add dishes to their cart and proceed to a hassle-free checkout, while managers can oversee the entire ordering process.

**Managerial Control:**

Managers have the power to curate and update the menu, ensuring an ever-evolving and enticing culinary experience for customers.

**Feature added for Final:** adding and retrieving list of reviews from customers about restaurant

Angular key technologies used in project**:** Material Design, DI, RxJS, localstorage usage, PWA for caching, HttpInterceptor for auth(JWT token),   
Django key technologies used in project: REST framework, function based views, class based views(mixins), sqlite3 as database, field validation, serializers