Minwoo Kwak

Cloning **Doordash**

Tools:

- Angular
- Node.js
- Express
- MongoDB

Packages:

- express: express server
- nodemon: dev dependency, boost DX
- mongoose: MongoDB ORM
- dotenv: store encrypted data/password
- bcrypt: encrypt user password
- jsonwebtoken: use JWT for user authentication
- cors: solve cross-origin issues

Feature:

1) User Admin/Auth - Users shall be able to sign up, log in, and log out.
    a) Email must be unique for each user.
    b) JWT provided upon login(lasts for 1 hour), saved on the browser.
    c) Logged in users shall maintain their status throughout the whole app.
    d) Token auth is necessary to place orders.

2) Main Page - Users shall be able to browse different restaurants.
    a) Restaurants can be filtered by rating, category, delivery fee, etc.
    b) Search bar to find restaurants by name and their food tags.
    c) Users do not have to be logged in to browse the main page.
    d) Authenticated users can navigate to Orders page.

3) Restaurant Page - Users shall be able to see a detailed menu for each restaurant.
    a) Each restaurant has its page with more detailed information and a menu.
    b) All Users shall be able to browse the menu.
    c) Clicking on a dish will open up a window to place an order.
        i) To place an order, users must be authenticated first (logged in).
        ii) Quantity can be specified when placing an order.
    d) Authenticated users can checkout if there are one or more items in the cart.
        i) Cart component displayed on the right side of the restaurant page.

4) Orders/Delivery Page - Users shall be able to track delivery progress and view order history.
    a) If a user placed an order (that hadn’t been delivered yet), it will get displayed on this page with data regarding the delivery.
        i) Deliverer info => name, rating, car type, ETA.

        ii) Order info => items, total price, address.
    b) Users shall be able to see past orders history.
        i) If there have been no orders yet, it will display a message to encourage first order.
    c) Delivery status will be either 1) resolved manually by an admin or 2) use setTimeOut() to denote delivery time and resolve status after time out.

Side Notes:

- For simplicity, I have skipped the Checkout Page (delivery options, payment etc). Clicking the checkout button will directly place an order to the specified user address, without further authentication or payment.
- There will be a pool of Delivery persons, which will get randomly assigned upon an order from users.

Models:

- User/Customer
    - Name: string
    - Address: addressId
    - Phone: string/number
    - Email: string
    - Password: string minLength=4
    - Orders: [orderId]
    - Cart: cartId
    - Address: addressId
- Deliverer
    - Name: string
    - Rating: number
    - CarType: string
    - ETA: number
    - Phone: string/number
- Restaurant
    - Name: string
    - Menu: [DishId]
    - DeliveryTime: number
    - Rating: number
    - Reviews: [reviewId]
    - DeliveryFee: number
    - Category: string (“burgers” | “asian” | “fastfood”)
    - Tags: [string]
- Dish
    - Name: string
    - Description: string
    - Ingredients: [string]

    - Price: number
    - Tags: [string]
- CartItem
    - Dish: dishId
    - Quantity: number
    - Cart: cartId
- Cart
    - Items: [cartItemId]
    - Userinfo: userId
    - deliveryAddress: addressId
    - totalPrice: number
        - Price is calculated by foodPrice * quantity
- Order
    - Items: [cartItemId]
    - Userinfo: userId
    - deliveryAddress: addressId
    - totalPrice: number
    - Status: string (“pending | delivered”)
    - deliveryInfo: delivererId
- Review
    - Restaurant: RestaurantId
    - ReviewText: string
- Address
    - Street: string
    - City: string
    - Zip Code: number optional
    - Country: string
    - User: userId