We want to create a website for a restaurant, the purpose of this is to order from a restaurant through a specified list of menu items and to handle everything including payment and delivery and an admin portal for the restaurant’s owner.

**The Web portal will include the following pages**

**Register Page**

To enter the website. The register page contains:

* Fancy background image
* Login form:
  + Logo
  + Username (textbox)
  + Name (textbox)
  + Password (hashed textbox)
  + Role (dropdown)
  + Register button

**Login Page**

To secure the web portal, we need to have a login page to let authorized users access it. The login page contains:

* Fancy background image
* Login form:
  + Logo
  + Username (textbox)
  + Password (hashed textbox)
  + Login button

**Home Page**

The main page will include:

* Header menu:
  + Home
  + Admin Panel (Menu Items, My Orders, All Orders)
  + Shopping Cart
  + Logout
  + Note: Admin Panel disappears for customers and is replaced by My Orders
* Body section:
  + Search bar
  + Menu items that you can add to cart
  + Filter according to category
  + Sort according to price and name
* Footer section

**Menu Item**

This section is for the restaurant owner (admin) to manage the menu items.

1. **Menu Items List**
   * Displays a list of all menu items.
   * Includes options to add a new item, edit existing items, and delete items.
   * Each item shows basic information like name, price, and category.
   * Pagination controls to navigate through the list if there are many items and sorting according to price and name.
2. **Menu Item Upsert**
   * A form to add a new menu item or update an existing item.
   * Form fields include:
     + Name (textbox)
     + Description (textarea)
     + Special Tag (textbox)
     + Category (dropdown)
     + Price (textbox)
     + Image upload (file input)
   * Save button to save the changes.
3. **Menu Item Details**
   * Displays detailed information about a specific menu item.
   * Includes fields like name, description, price, category, and image.
   * Option to add to cart or go back to home.

**Order**

This section is for viewing and managing orders.

1. **All Orders**
   * Displays a list of all orders placed by customers.
   * Each order shows basic details like order number, customer name, customer phone, total amount, total items, date and status.
   * Options to view order details, update status, and delete orders.
   * Pagination controls to navigate through the list if there are many items and search bar and filtering according to status.
2. **My Orders**
   * Displays a list of orders placed by the logged-in customer.
   * Each order shows details like order number, date, total amount, and status.
   * Options to view order details and track delivery status.
3. **Order Confirmed**
   * A confirmation page displayed after an order is successfully placed.
   * Shows order number.
4. **Order Details**
   * Displays detailed information about a specific order.
   * Includes customer details(name, email, phone), list of items ordered with quantities and prices, total amount, and delivery details and status.
   * Option for the admin to update order status (e.g., processing, out for delivery, delivered).

**Shopping Cart**

This section allows customers to view and manage their shopping cart.

1. **Cart Summary**
   * Displays a list of items added to the cart.
   * Each item shows name, quantity, price, and total cost.
   * Options to update quantities or remove items from the cart.
   * Shows the total amount for the items in the cart.
2. **Pickup Details**
   * A form for customers to provide pickup details.
   * Form fields include:
     + Name (textbox)
     + Phone number (textbox)
     + Email (textbox)
     + Pickup date and time (date and time picker)
   * Option to choose between delivery and pickup.
3. **Checkout Button**
   * A button to proceed to the payment page.

**Payment**

This section handles the payment process.

1. **Payment Details**
   * A form for customers to enter payment information.
   * Form fields include:
     + Card number (textbox)
     + Expiry date (textbox)
     + CVV (textbox)
     + Billing address (textbox)
   * Option to apply discount codes or gift cards.
2. **Payment Summary**
   * Displays a summary of the order including items, quantities, total amount, and any discounts applied.
3. **Place Order Button**
   * A button to complete the payment and place the order.
4. **Payment Confirmation**
   * A confirmation page displayed after successful payment.
   * Shows payment confirmation number, order details, and estimated delivery or pickup time.

**Technologies used in this project**

* Frontend
  + Reactjs & Redux
  + Bootstrap
* Backend
  + .Net 8 API
  + C#
  + Entity Framework
* Cloud
  + Microsoft Azure

**Code Design**

* Models
  + User (Id, Username, Email, PasswordHash, UserRoleId, Name) Using Identity
  + UserRole (Id,Name) Using Identity
  + MenuItem (Id, Name, Description, SpecialTag, Category, Price, Image)
  + OrderDetails (OrderDetailId, OrderHeaderId, MenuItemId, Quantity, ItemName, Price)
  + OrderHeaders (OrderHeaderId, PickupName, PickupPhoneNumber, PickupEmail, ApplicationUserId, OrderTotal, OrderDate, StripePaymentIntentId, Status, TotalItems)
  + ShoppingCarts (Id, UserId)
  + CartItems (Id, MenuItemId, Quantity, ShoppingCartId)
* API Controllers
  + Authentication (Register, Login Methods)
  + MenuItem (GetList, GetById, Add, update & delete methods)
  + Order (GetList, GetById, Add, update methods)
  + ShoppingCart (GetById, Add, update methods)
  + Payment (Add method)
* DB Context
  + Add entityframerowrk support
  + Use Identity
  + Migrate Models to database
  + On model creating override the function to add Admin
* Functionalities
  + API methods should send & received DTO objects and not the actual models, so you need to have a DTO objects and a mapper for those DTOs
  + User Password should be encrypted(hashed) while saving in database, so you need to hash the user password before saving into database and validate password when loading user from the database

**UI Design**

* Use what you have learned in terms of design and page rendering using reactjs & redux
* Make sure that the design is responsive and user-friendly
* Use all needed reactjs libraries and tool to finalize the requested features

**Deployment**

* Deploy App and Api on Microsoft Azure

API Endpoints:

Auth/Register: POST

Auth/Login: POST

AuthTest: GET

AuthTest/{id}: GET

MenuItem: GET

MenuItem: POST

MenuItem/{id}: GET

MenuItem/{id}: PUT

MenuItem/{id}: DELETE

Order: GET

Order: PUT

Order/{id}: GET

Order/{id}: PUT

Payment: POST

ShoppingCart: GET

ShoppingCart: POST