How to PLAN a New App.

# Customer – Fast React Pizza Co.

* The pizza guys from our first project
* They want a simple way of allowing customers to **order pizza and get them delivered** to their home
* We are hired to build the application front-end

### From the ‘Thinking in React’ lesson, we had the initial 3 steps

### Break the desired UI into Components

### Build a static version (no state yet)

### Think about state management + data flow

### These steps work for small applications, but for a big, real-world application, we need to expand on the process

STEPS

## Gather application requirements and features

## Divide the application into pages

### Think about the **overall** and **page-level** UI

### Break the desired UI into components

### Design and build a static version (no state yet)

## Divide the application into feature categories

### Think about state management + data flow

## Decide which libraries we want to use (technology decisions)

Steps 1 by 1

# Project Requirements from the Business

* Very simple application, where users can order **one or more pizza from a menu**
* Requires **no user account and no login**: users just input their names before using the app
* The pizza menu can change, so it should be **loaded from an API** (already done – backend done)
* Users can add multiple pizzas to a **cart** before ordering
* Ordering requires
  + User’s name
  + Phone number
  + Address
* If possible, **GPS location** should also be provided, to make delivery easier
* Users should be able to **mark their orders as priority**, for an additional 20% of the cart price
* Orders are made by **sending a POST request** with the order data to the API
  + User data
  + Selected pizzas
* Payments are made on delivery, **no payment processing**
* Each order will get a **unique ID** that should be displayed, so the **user can later look up their order** based on the ID
  + API call based on ID
* The order can be **marked as “priority” even after the order has been placed**

# + 3. Features + Pages

* From the Requirements we can develop further and plan out what pages and features we need to implement
* We first derive the **Features** 
  + User
    - All the features that are related to the user
  + Menu
    - All the features that are related to the menu
    - Display
    - Update
  + Cart
    - Where the user will be able to add to cart
    - Update the quantity
  + Order
    - Place a new order
    - Looking up an existing order
* From the Features, we can also understand the **Pages** that we are going to implement
  + Homepage -- /
  + Pizza Menu -- /menu
  + Cart -- /cart
  + Placing a new order -- /order/new
  + Looking an order -- /order/:orderId

# 4. State Management + Technology **Decisions**

* State Slices
  + User slice
  + Menu slice
  + Cart slice
  + Order slice
* Types of **State**
  + User 🡪 Global UI State (no accounts, no stays in app)
  + Menu 🡪 Global Remote State
  + Cart 🡪 Global UI State
  + Order 🡪 Global Remote State

### Tech Decisions

* Routing
  + **React Router** or React Query
    - Standard for React SPAs
* Styling
  + **Tailwind Css**
    - Popular way to style the application
* State Management
  + **Remote State Management**
    - **React Router**
      * New data fetching capabilities since v6.4+
  + **UI State Management**
    - **Redux**

Professional File Structure

* The file structure is something that each developer does as he finds useful
* On this project, we will use the **Feature Based** Structure
  + One **Features** folder
  + Each feature will have it’s own folder
  + There are also components that don’t belong to any of the features
    - A **UI folder** for reusable components
      * Buttons
      * UI
      * Modals
  + The **Services** folder
    - For interacting with the **API**
  + The **Utils** folder
    - For **helper functions**
    - Function that don’t keep state
    - Date or number manipulation