# Exam Preparation – Furniture System

## Starting with a Mock

Before beginning work on the code, look at the shape of your data, and prepare a UI mock.

### Data

Analyze the API endpoints and extract a couple entries to be used in testing. You may test everything in Postman or another tool.

### UI Mock

Prepare a prototype on paper, using an image editor or with HTML and CSS.

## Component Hierarchy

Follow Single Responsibility principle.

Main components: HomePage, RegisterPage, LoginPage, CreatePage, DetailsPage, ProfilePage

Sub components: header, stats, furniture card, paginated view, pagination, search form, login form, register form, create form, review card, review form, details card

## Prepare Project

Install libraries and setup folder structure. Install redux, react-redux, redux-thunk, react-router, react-router-dom. If notifications are desired, install jquery, toastr.

## Static React Version

Place HTML into components that render data using only props (no interactivity). Begin with placeholder page components, to get the routing working, then implement each page.

## Representation of UI State

Analyze all data inside your app and think about what parts of it might change. Only store the data that cannot be passed in as props and cannot be computed from props and state.

**State**: current page, search string, create form input, create form validation, review input, review rating

## State Store

Identify where to keep your state – place data inside your Redux store, UI-only state should remain in the components.

## Inverse Data Flow

Make components deeper in the hierarchy talk to their parents higher up. Either use Redux, pass down change/notify handlers or create an external observer.