

Batch T-6

Practical No. 6

Title of Assignment - Study and implementation of ReactJs

Student Name: Harshavardhan Anil Bamane

Student PRN: 22510112

Problem Statement 1: To-do List Application with State Management

Build a simple To-do List application that allows users to add, remove, and mark tasks as completed. The app should have a responsive user interface that works well on both desktop and mobile devices. Users should be able to input a task, which will be added to the list. Each task should have a checkbox to mark it as completed. Users can remove tasks from the list. Use React hooks like useState for state management. Add options to filter tasks based on completion status (e.g., all, completed, active). Break down the Todo List into multiple components, like TaskList, TaskItem, AddTaskForm, etc. The app should display appropriate error messages. The candidate should use appropriate error handling and validation to ensure that the application is robust and user-friendly.

To-Do List

Wonderful! Yippe you are a productive person now!

ADD

ALL

ACTIVE

COMPLETED

Study

Assignments

Singing

To-Do List

Wonderful! Yippe you are a productive person now!



Add a new task

ADD

ALL

ACTIVE

COMPLETED

Assignments

Singing

To-Do List

Wonderful! Yippe you are a productive person now!



Add a new task

ADD

ALL

ACTIVE

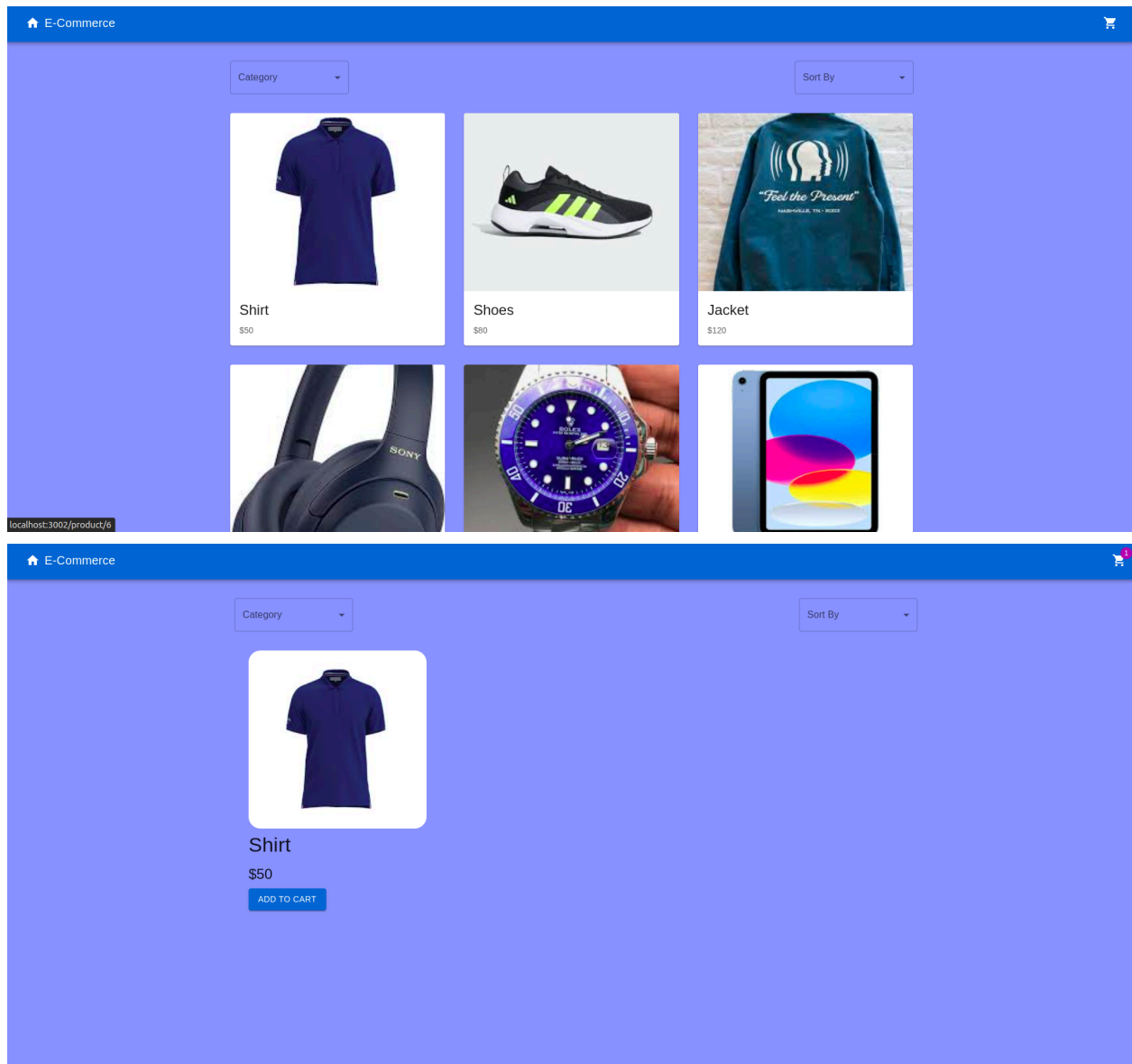
COMPLETED

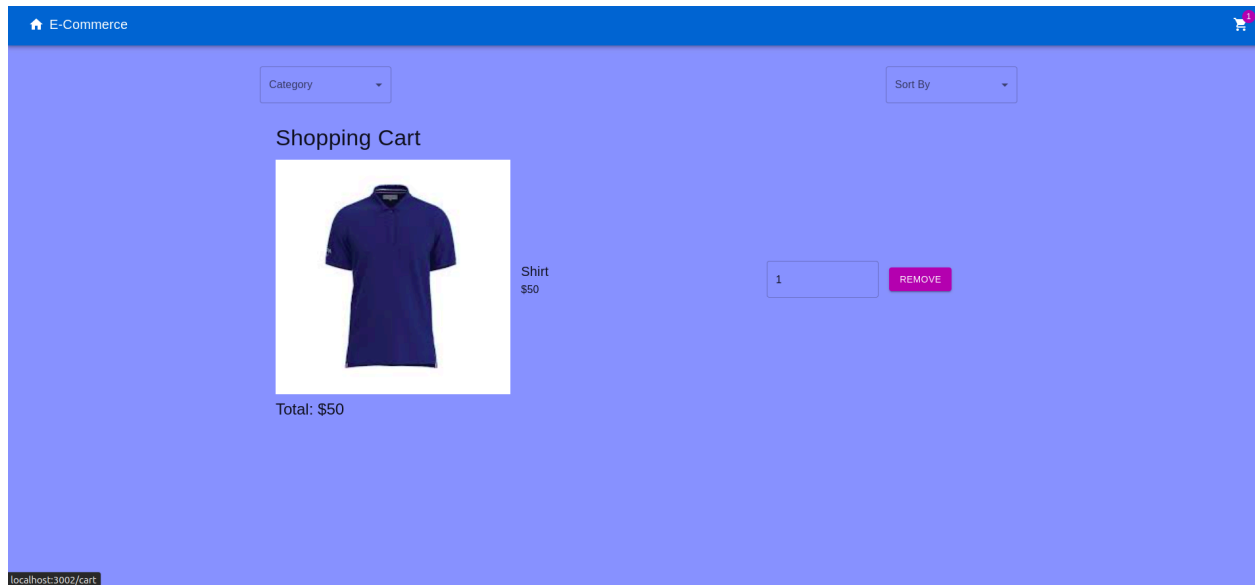
Study

Problem Statement 2: Simple E-commerce Cart System

Requirements:

Develop a small e-commerce application where users can browse products and add them to a shopping cart. Display a list of products with details like price, name, and image. Users can add products to a shopping cart. The cart page shows all selected products and their total price. Users can remove items from the cart or adjust the quantity. Add filtering options (e.g., by price, category) and sorting (e.g., low-to-high, high-to-low). Add routing to navigate between different pages, like the product list, cart, and checkout pages. On clicking a product, navigate to a detailed view of that product.

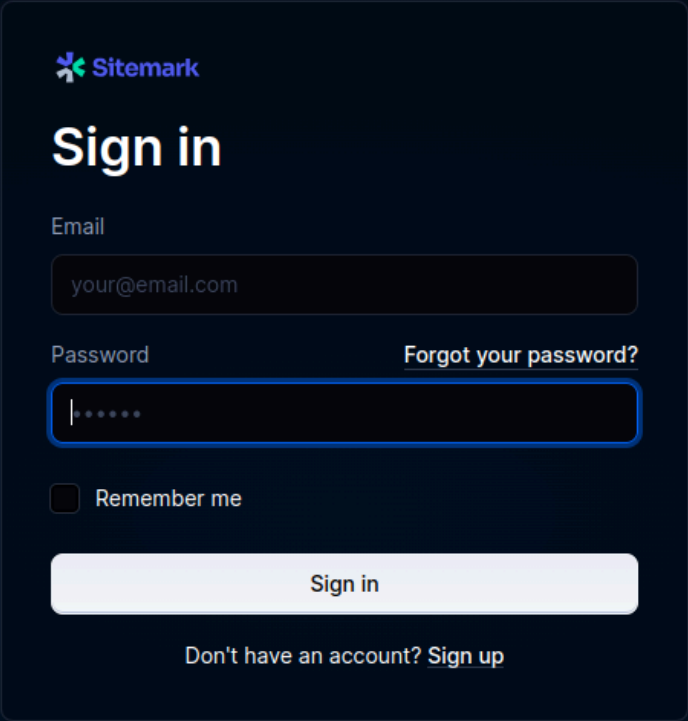





Problem Statement 3: User Authentication with React Context API

Requirements

Create a login form that checks for hardcoded user credentials. If authenticated, the user can see a personalized dashboard. Allow the user to log out, which resets the context state. Use React Router for navigation between login and dashboard pages. Protect certain routes (e.g., dashboard) so that only authenticated users can access them.



The image shows a 'Sign in' form for 'Sitemark' on a dark blue background. The form is centered and contains the following elements: the Sitemark logo, the title 'Sign in', an email input field with the placeholder 'your@email.com', a password input field with a blue border and masked characters, a 'Remember me' checkbox, a 'Sign in' button, and a link for users without an account.

 Sitemark

Sign in

Email

your@email.com

Password [Forgot your password?](#)

|

☐ Remember me

Sign in

Don't have an account? [Sign up](#)

