

SWE 4638
Web and Mobile Application Development
Lab No: 2

Task 1: React Lab Task: Task Manager Tracker

Objective

Build a ReactJs application that keeps track of the Task completion status. The website should be dynamic based on the number of tasks available. The application should have the following.

- ☐ Display a list of tasks with their titles and completion status
- ☐ Show completed tasks differently from incomplete tasks (styling/strikethrough)
- ☐ Display a counter showing total tasks and completed tasks



Interactive Task Manager

Click tasks to toggle completion, add new tasks, and manage your to-do list!

3

Total Tasks

1

Completed

2

Remaining

Add Random Task

Clear Completed (Available)

Learn React basics

○ Pending

Remove

~~Practice useState hook~~

✓ Completed

Remove

Build first React app

○ Pending

Remove

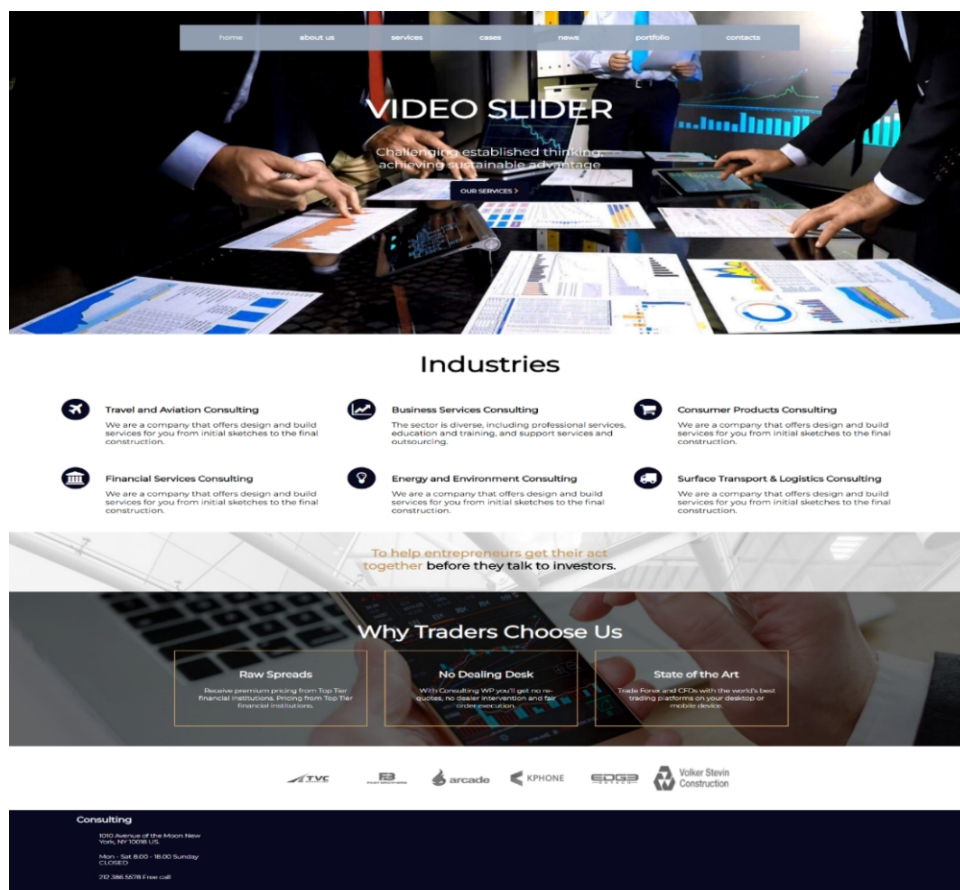
A sample list of task is given. These tasks are used as state of the component.

```
[  
  { id: 1, title: "Learn React basics", completed: false },  
  { id: 2, title: "Practice useState hook", completed: true },  
  { id: 3, title: "Build first React app", completed: false },  
]
```

Task 2:

Using the website that you created from lab 1, transform the html and css landing page into a ReactJS web application. The following components have to be included.

- MainComponent
- NavComponent
- HeroComponent
- Industries
- Industry
- EntrepreneursComponent
- TradersMain
- Trader
- Partners
- FooterComponent



Task 3: Personal Profile Card

Objective

Create a simple personal profile component that displays information about a person, including their image.

Requirements

1. Create a `ProfileCard` component that displays:
 - A profile image
 - Name
 - Job title
 - Company
 - A short bio
 - Skills list (displayed horizontally)
 - Contact button (that logs a message to console when clicked)
2. The component should accept the following props:
 - `name` (string)
 - `jobTitle` (string)
 - `company` (string)
 - `bio` (string)
 - `skills` (array of strings)
 - `imageUrl` (string)
3. Style your component with CSS to create a professional-looking profile card.

In your main `App.js` file, render at least 3 different `ProfileCard` components with different data.