## Todo List App: a case study



Reflection **Planning** Development Objective: Upon discovering Redux in my CareerFoundry curriculum, I was eager to delve deeper into its capabilities. I decided to rebuild a vanilla JavaScript todo-list app from earlier in the course, using React Conclusion: combined with Redux for Rebuilding the todo-list app advanced state with React and Redux was a management. My aim was significant milestone in my to enhance the app's coding journey. The project scalability and functionality not only introduced me to through Redux's advanced state predictable state management techniques but Structure: management. also enhanced my technical I initiated the project with skills and independent Create React App (CRA), problem-solving capabilities. which marked a significant Moving forward, I plan to improvement in my terminal improve the app with new skills. Integrating Redux features like card-like views involved defining actions for and multiple list managing todo items, management. creating reducers for state changes, and connecting these components to the Redux store. This setup allowed for efficient and predictable state **Todo List:** management across the app. Add Item: grocery shopping X weed flower bed X mow the yard  $\chi$ Design: walk the dog  $\chi$ I then created a very simple but intuitive UI using React-Bootstrap. call mom X Conceptualization: The goal was to transform plan dinner X my basic JavaScript todolist app into a more robust and scalable application by integrating Redux. With Redux, I could implement efficient state management for adding, Designed and developed by: Leanne Duyck toggling, and deleting todo items. The interface would be simple, and would use Bootstrap-React. Deployment: The initial deployment failed due to incorrect start instructions in the package.json file. With help from ChatGPT, I updated the start instructions and successfully deployed the

app. This troubleshooting

experience was a valuable

learning opportunity.

ReaCt