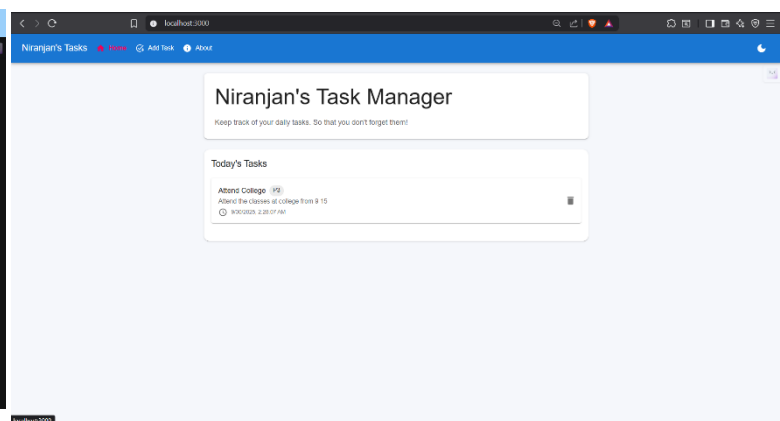
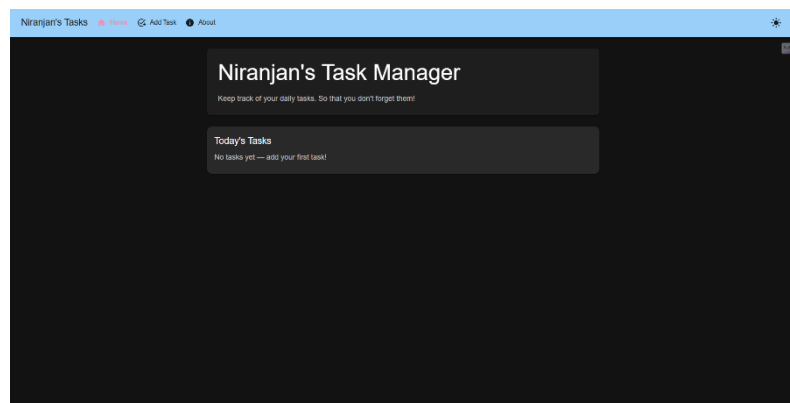
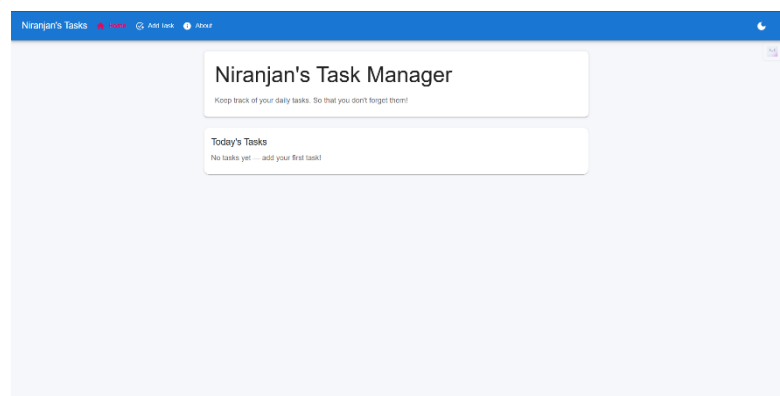
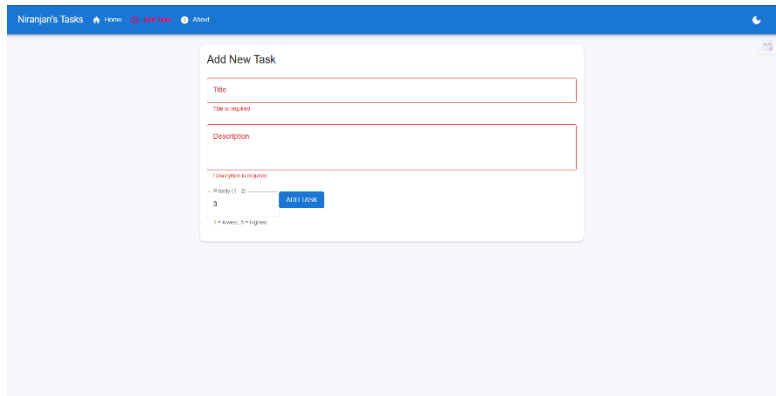


## AeroAspire -SDE Intern Training

NIRANJAN C N

Week2-Day4-Oct3

Add routing: Home, Add Task, About; navigation bar, Save tasks to localStorage; load on mount; polish UI; icons; theming



- Added a navbar for navigation around pages
  - Added icons for each button
  - Added a separate page for adding tasks
  - A priority can be set for each task
  - Task can be deleted and stored
  - The task entered is stored locally and loaded on mount
  - The ui is a bit polished with hover effects and buttons
  - A toggle between dark and light theme is available
  - The logic remains the same as previous day's task but the ui and the presentation has improved
- Questions

1. Describe how client-side routing works (history API or hash routing).
  - a. History API-
    - i. React router uses browser history API for manipulating the url and search history
    - ii. URLs update without a page reload so that the forward and backward navigation takes place smoothly
  - b. Hash routing-
    - i. Navigation occurs completely in client in hash routing
    - ii. Hash routing uses window.location.hash
2. What happens when you navigate: how React Router matches route and renders components.
  - a. When location changes of a particular component the react router needs a new url
  - b. It matches the url to the route specified and finds the correct path.
  - c. And after that it renders the corresponding react components
  - d. Nested routes are used to show both the parent and child components at the same time
3. How to pass params or query params; nested routes.
  - a. Path params- /task/:id, the param is available in the components through hooks like useParams.
  - b. Query params- /task?id=999, We can access them like useSearchParams
  - c. Nested routes are used to show both the parent and child components at the same time
4. What is the flow: writing to localStorage → reading on app startup?
  - a. Writing- We use localStorage.setItem(key,value) after any state update
  - b. Reading- On app startup use localStorage.getItem(key) to read the data input
  - c. Initialize state- On startup state is obtained from local storage if existing else load defaults
5. How do you sync state with localStorage safely (e.g. updates, JSON parse/stringify)?
  - a. We can stringify objects/arrays before saving (JSON.stringify)
  - b. (JSON.parse) to read input and check for errors in input
  - c. Sync state on any updates using effects/hooks
6. What performance / size concerns with storing too much in localStorage?
  - a. Browsers usually allow a limit of 5Mb for storage
  - b. Too much data can lead to slow read and write operations
  - c. It might also lead to errors when the limit is reached
  - d. We should not store sensitive data and large data