

Personal Task Tracker - Intern Pre-Hire Assignment

Assignment Overview

Build a simple personal task management application using **React**. This assignment will evaluate your React skills, component organization, and problem-solving approach.

- **Estimated Time:** 3–4 hours
 - **Tech Stack:** React.js (Frontend only)
-

Project Requirements

1. Simple Login

- Basic login form with a username input
 - No real authentication — just store the username in `localStorage`
 - Redirect to task dashboard after "Login"
-

2. Task Management

- **Add Task:** Form with title (required) and description (optional)
- **Edit Task:** Edit inline or via modal
- **Delete Task:** Prompt confirmation before deletion
- **Toggle Complete:** Checkbox or button to mark task as completed/pending

3. Task Display

- Show task title, description, and completion status
- Display creation date/time
- Visual distinction between completed and pending tasks

4. Task Filtering

- Tabs or buttons for:
 - **All**
 - **Completed**
 - **Pending**
- Show task count for each filter

5. Data Persistence

- Use `localStorage` to save tasks
- Tasks should persist after page refresh

Technical Requirements

- Use **React functional components** with **hooks**
- **Responsive design** (mobile + desktop)

- **Clean, readable code** with proper component structure
 - Basic styling using **CSS**, **SCSS**, or **styled-components**
 - No external state management libraries (use React's built-in state)
-

What We're Evaluating

Criteria	Weight	What We Look For
React Skills	40%	Hooks, component organization, props handling
Code Quality	25%	Clean code, naming conventions, file structure
UI/UX	20%	User-friendly interface, responsive design, basic styling
Functionality	15%	All features working correctly, edge case handling

Project Structure Suggestion

pgsql

```
task-tracker/  
├─ public/  
│   └─ index.html  
├─ src/  
│   ├─ components/  
│   │   ├─ Login.js  
│   │   ├─ TaskForm.js  
│   │   ├─ TaskItem.js  
│   │   ├─ TaskList.js  
│   │   └─ TaskFilter.js  
│   ├─ utils/  
│   │   └─ localStorage.js  
│   ├─ styles/  
│   │   └─ App.css  
│   └─ App.js
```

```
|   └─ index.js
|─ README.md
└─ package.json
```

🌟 Bonus Points (Optional)

- Search functionality
 - Task priority levels
 - Due dates for tasks
 - Smooth animations/transitions
 - Dark mode toggle
 - Task categories/tags
-

📦 Submission Requirements

1. GitHub Repository

- Public repo with a **descriptive name**
- Clear commit history showing development progress
- Include all source code

2. README.md

markdown

```
# Personal Task Tracker
```

```
## 📖 Description
[Brief description of your app]
```

🚀 Features

- [List key features you implemented]

🛠️ Setup Instructions

1. Clone the repository
2. Run `npm install`
3. Run `npm start`
4. Open http://localhost:3000

📦 Technologies Used

- React.js
- [Any other libraries you used]

🔗 Live Demo

[Link to deployed application]

🖼️ Screenshots

[Include 1-2 screenshots of your app]

3. Live Demo

- Deploy using **GitHub Pages**, **Netlify**, or **Vercel**
- Ensure the deployed version works properly
- Include the **live URL** in your README



Getting Started

Setup Instructions

Create your React app

bash

```
npx create-react-app task-tracker
```

1. Clean up unnecessary files
2. Plan your component structure
3. Start with the login component
4. Build task management features
5. Add filtering functionality
6. Style your application
7. Test thoroughly
8. Deploy and document



Sample Data for Testing

js

```
const sampleTasks = [  
  {  
    id: 1,  
    title: "Complete React assignment",  
    description: "Build a task tracker application",  
    completed: false,  
    createdAt: "2024-01-15T10:00:00Z"  
  },  
  {  
    id: 2,  
    title: "Review JavaScript concepts",  
    description: "Go through ES6+ features",  
    completed: true,  
    createdAt: "2024-01-14T15:30:00Z"  
  }  
];
```



Evaluation Rubric

Rating	Description
Excellent (4/4)	All features work, clean code, responsive UI, good UX, proper React usage
Good (3/4)	Most features work, mostly clean code, decent UI and responsiveness
Satisfactory (2/4)	Core features functional, organization/styling may be lacking
Needs Improvement (1/4)	Missing key features, messy code, poor UX



Common Pitfalls to Avoid

Don't:

- Over-engineer the solution
- Use complex state management
- Spend too much time on advanced styling
- Skip README documentation
- Forget to test the deployed version



Do:

- Keep it simple and functional
 - Write clean, readable code
 - Test all features thoroughly
 - Document your work clearly
 - Focus on core requirements first
-

Submission Process

1. Complete the assignment within the suggested timeframe
2. Push your code to GitHub
3. Deploy your application
4. Send us:
 - GitHub repository URL
 - Live demo URL
 - Any additional notes or comments

? Questions?

If you have any clarifying questions, feel free to reach out. We're looking to see your **problem-solving** and **React fundamentals** — not perfection.

Good luck! We're excited to see what you build! 🚀