Setup Instructions

- **Prerequisites**:
  - List any software or libraries needed (e.g., Node.js, Python, etc.).
- **Installation Steps**:
  1. Clone the repository:

```bash
git clone [repository-url]
```

2. Navigate to the project directory:

```bash
cd [project-name]
```

3. Install dependencies:

```bash
[package-manager] install
```

- **Running the Project**:
  - Instructions on how to run the project locally (e.g., `npm start`, `python app.py`, etc.).

Slack and LLM Setup Guidance

- **Configuring Slack**:
  - Steps to create a Slack app and obtain API tokens.
  - Instructions on setting necessary permissions and scopes.
- **Integrating the LLM**:
  - Steps to set up the language model.
  - Necessary API keys and configuration settings.

Slack Integration:

- Go to Slack → Add Apps → Search for **Incoming Webhooks**.
- Select your desired channel and create a webhook.
- Copy the generated webhook URL.
- Add the following to your `.env`:

```env
SLACK_WEBHOOK_URL=https://hooks.slack.com/services/...
```

✅ 3. Design/Architecture Decisions

Frontend:

- Built with **React** using `useState`.
- Allows adding and marking tasks as completed.
- Sends data to backend for summarization and displays results.

Backend:

- Built with **Express.js** in **Node.js**.
- Exposes a POST `/summarize` endpoint.
- Uses **OpenAI API** to generate a task summary.
- Posts summary to **Slack** via webhook.

✅ 4. (Optional) Deployed URL

Frontend Hosting Options:

- Vercel
- Netlify
- Firebase Hosting

Backend Hosting Options:

- Render
- Railway
- Heroku