

# Phase 2 Guided Project Guidelines

In this guided project, we'll be building an Evernote-like application with React.

Your goal will be to complete the baseline deliverables using the provided starter code and then make this project your own by building out unique stretch goals. Some suggestions are listed below, but try to be creative!

## Requirements

- Complete all of the deliverables
- Build out at least 1 stretch goal feature

## Setup

To get started, fork and clone this repository.

The codebase is split up into a JSON Server API backend and a React frontend.

## Frontend

To setup and run your frontend:

```
$ npm install
$ npm start
```

This React app will be running on <http://localhost:4000>.

## Backend

First, set up your database by running:

```
$ npm run seed
```

This will seed some starter data for you in the `db/db.json` file. Any time you want to reset your database to its original state, just run `npm run seed` again to overwrite your data with some fresh seed data.

To run `json-server`, run:

```
$ npm run server
```



Your backend API will be running on `http://localhost:3000`.

## User ID

The seed file should create one user for you, so your default `userId` should be `1`. You can check the `db/db.json` file to make sure.

## Routes

Method	Route	Headers	Body
GET	<code>/users</code>		
GET	<code>/notes</code>		
POST	<code>/notes</code>	<code>'Content-Type': 'application/json' ; 'Accept': 'application/json'</code>	title, body, userId
PATCH	<code>/notes/:id</code>	<code>'Content-Type': 'application/json' ; 'Accept': 'application/json'</code>	title, body, userId

### Tips:

- Test out your routes with [Postman](https://www.getpostman.com/) [\(https://www.getpostman.com/\)](https://www.getpostman.com/) to see how they work and what they return.

## Provided Code

- All CSS styles are provided for you.
- Many components are provided for you, but most are not completely functional. It is your job to read the code and figure out how to incorporate it into your app.

## Deliverables

Look at the gif below to see how the app should look and behave. These are the baseline deliverables you need to complete:

## Viewing and Displaying Notes

- [ ] Display all notes in the left sidebar.
- [ ] Displayed sidebar notes should show the title and a truncated body.
- [ ] When a note from the sidebar is clicked, display its contents in the right panel.

## Garbage Collectors' Flatnote

Search Notes

conspergo.

Cur sordeo ver. U...

Audacia

antea

balbus

spiculum.

Summisse facere c...

Select a note.

## Filtering Notes

- [ ] Implement the filter to search through your notes list by title.

## Garbage Collectors' Flatnote

Search Notes

spiculum.

New edit

Utroque

tepesco.

Consequatur aggre...

Odit ea

Select a note.

## Creating Notes

- [ ] At the bottom of your left sidebar, show a **New** button.
  - [ ] Clicking **New** will create a new note via a **POST** request with some default title.
- NOTE:** You don't have to use any kind of `<form>` element for this deliverable;

 **Help**

an object with a default title and body text when the button is clicked. Make sure to check the [Routes](#) section of this README to determine what data you need in the body of your request.

- [ ] This new note should appear in the sidebar.



## Editing Notes

- [ ] When displaying a note in the right panel, show an **Edit** button.
- [ ] Clicking the **Edit** button will allow the user to edit the title and body in the right panel.
- [ ] When in edit mode, also show a **Save** button which saves the note via a **PATCH** request.
- [ ] When in edit mode, also show a **Cancel** button which discards any changes and reverts back to displaying the note.
- [ ] Clicking a different note while in edit mode should discard your changes and display the new note instead.

## Garbage Collectors' Flatnote

Search Notes

Adipiscor  
conspargo.

Cur sordeo ver. U...

Audacia  
antea  
balbus  
criculum

Select a note.

## Stretch Goals

When you are finished with the *deliverables*, you can build out any new features that you want. This is your chance to be creative and make your project unique!

Some suggestions:

- Add the ability to filter by body, date created, date edited, etc.
- Sorting by date created, date edited, alphabetical, etc.
- Use `react-router` to create a multi-page app
- User signup & login
- Sharing notes with other users
- Rich text formatting
- Tagging
- Emailing notes