**readme**

**Overview**

This is a clone of the popular blogging network Medium. It implements the functionality to:

* Create and update posts
* follow users
* receive personalized content based on topic preferences
* edit reading history
* update and reset login information
* comment and reply to comments
* like both posts and comments

**More specifics**

The App uses React’s Context API to pass user auth details to different components, passing in props using higher order functions (ContextWrapper.js) for props outside the render method, and UserConsumer for within. Several components also implement Hooks.

Sessions are handled with Express sessions and email authentication with JWT.

**Third Party**

The project users PassportJS for authenticating users and Nodemailer for both validating login details and also resetting the user’s password, where emails with a link are sent by way of smtp.gmail.com . API calls are done with Axios and file uploads are handled with Multer.

**Main files**

The two main parent files are UserContext.js and App.js.

**Live Version**

You can see a live working version hosted on Heroku here:

**Why build it?**

This is intended as a portfolio project.

**Usage**

The app already contains post and user data

I do not have enough email addresses and thus some of the authors share the same email address

**Installation**

After downloading:

* npm install

then:

* Production: npm run concurrently
* Dev mode with Nodemon: npm run devcon

**Case study**

**Motiviation**

I began with nothing more than a CRUD application allowing users to upload, edit and delete posts, however, by copying some of the functionality present on the popular blogging platform Medium, it provided an opportunity to attempt to implement that functionality using my current knowledge and also develop it further. I built this project after having completed the Quiz App, also in my protfolio.

**Choices made**

My main goal was to build the site using common frameworks and third party software, in order to gain familiarity with tools frequently used int industry. I chose:

PassportJS for login authentication, being widely used and allowing in future to implement Oauth.

JSON web tokens

As the application needs access to user data to perform API calls throughout the program I decided to try using React Context to make the data available globally avoiding so-called “props driling”.

**Challenges**

My original plan had been to upload the app to Heroku, however, after having tried, several issues became apparent. A “memory quota” error suggests that the app is likely suffering a memory leak, and there was a persistent proxy error when navigating to pages. These require more knowledge and are something I am beginning to look into.

Another issue that needs to be addressed is that of keys in components within the app, which React states is necessary.

**Issues and next steps**

The next steps are to begin designing some basic unit tests for the app, an area I have begun to learn about recently.

It is also necessary to continue to make the site fully mobile responsive and add in keys to objects that are output.

**Quiz App Readme**

**Overview**

A full stack app that allows users to create their own quizzes and then run them

**Third Party**

Mongoose,

**Why build it?**

Selecting options while clicking through various screens seemed to be a model which is quite applicable to a range of app ideas I had, and thus I decided to begin with something that seemed manageable.

**Installation**

npm install

npm run bla

**Future Development**

Tests would be an initial improvement to the app. Aside from this, I would like to implement a login system and some kind of scoreboard functionality as well as stats that display average user results on quizzes,

# React Journalling App - Medium Clone

## Overview

This is a clone of the popular blogging network Medium. It implements the functionality to:

- Create and update posts

- follow users

- receive personalized content based on topic preferences

- edit reading history

- update and reset login information

- comment and reply to comments

- like both posts and comments

## Why build it?

This is intended as a portfolio project. What began as a full-stack MERN CRUD app, eventually grew to incorporate many of the features present on Medium's own site. This is in no way an attempt to rival the site but rather is my own personal aim to develop my own skills with the MERN stack by attempting to rebuild a well-known platform.

## Usage

Introduce an email into the sign up field and you will receive an email with a link (valid for a limited time) redirecting you to finish the sign up.

## More specifics

The App uses React’s Context API to pass user auth details to different components.

It passes in props to child components using a higher order function (ContextWrapper.js) for props outside the render method, and UserConsumer for within.

Several components also implement Hooks.

Sessions are handled with Express sessions and email authentication with JWT.

## Third Party

The project users PassportJS for authenticating users and Nodemailer for both validating login details and also resetting the user’s password.

Emails with a link are sent by way of smtp.gmail.com, API calls are done with Axios and file uploads are handled with Multer.

## Live Version

You can see a live working version hosted on Heroku here:

blbal link

## Installation

.

```bash

cd React-Jounalling-App

npm install

npm run concurrently

```

## License

[MIT](https://choosealicense.com/licenses/mit/)