

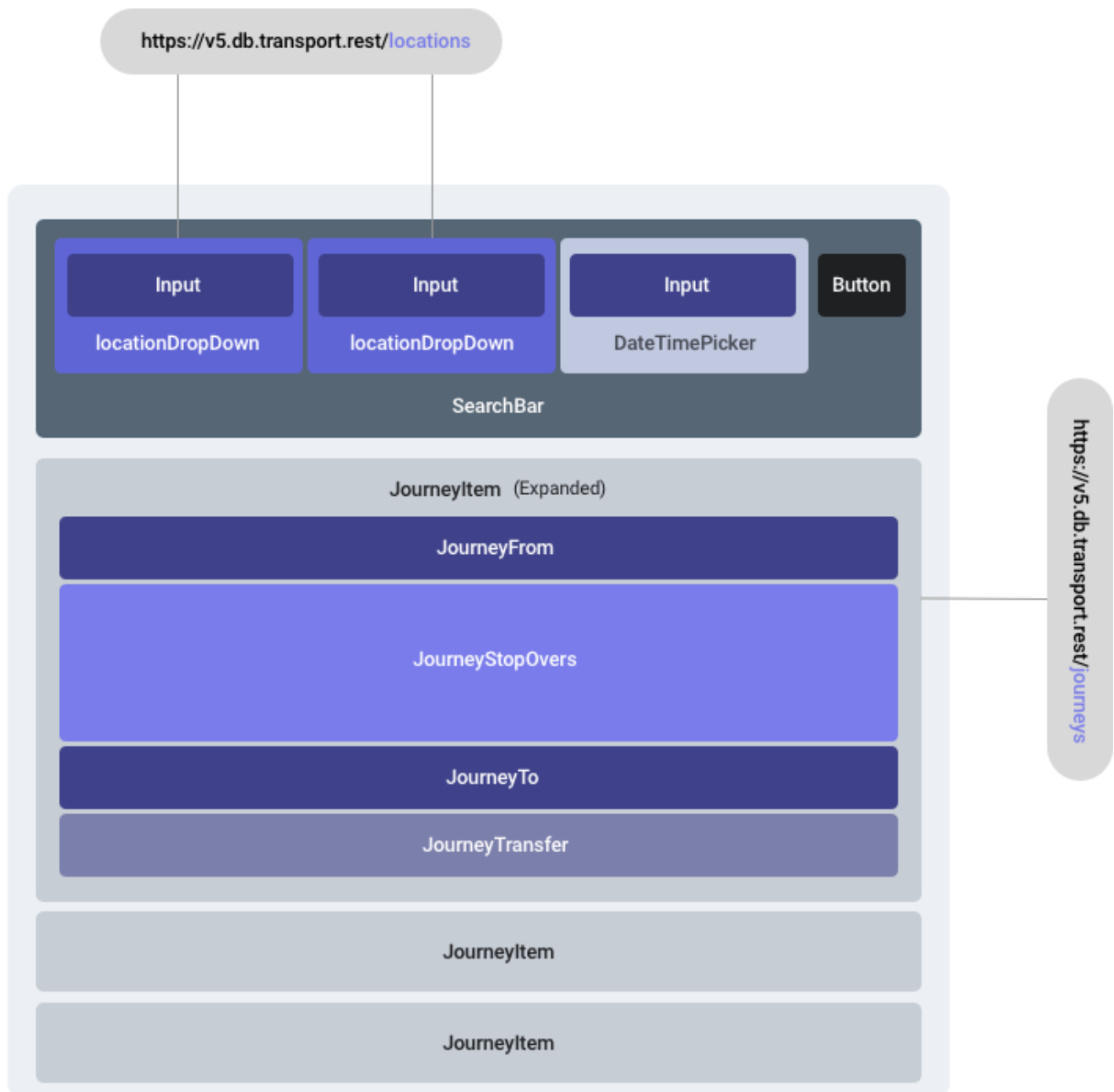
Deutsche Bahn Challenge

Trip planner application using Deutsche Bahn open API from <https://v5.db.transport.rest>

Libraries/ Tools used:

- Visual Studio Code
- Postman (For doing API Testing)
- Eslint
- Prettier
- Moment.js
- React Testing Library & Jest

Architecture and Components Breakdown



Execution Process:

- Understand different usability scenarios.
- Understand on which devices will the application be viewed.
- Sketching the draft design layout on a piece of paper.
- Identify and breakdown areas to reusable components.

- Create, style components and pages.
- Putting them all together and render using dynamic data.
- Structure code to be easily understandable to developers.
- Optimize performance and build sizes.
- Check functionality and write Unit Tests.

Features :

- Responsive design behaviour.
- User able to specify origin and destination along with a date and time.
- User to look for journeys by departure or arrival time.
- User not allowed to choose a date in the past.
- Skeleton loading for better user experience.
- Debounce for user inputs.
- Unit Test.

Features that could have been added:

- Buttons to look up for Earlier and Later Journeys.
- Add filters for all modes of transport
- Change number of results displayed on the page.
- Swap Origin and Destination values
- Show Point of Interests when clicking on Station / Stop names.
- Show a beautiful route on the Map.
- Link to save to your Calendar.
- Showing more detailed information about Trains
- There are places to improve. Still, I would not opt to do them all for a small problem domain like this one. As software engineers we need to find a balance.

Ready to deploy build

Just copy the files in the '**prod**' folder to the root directory of your host and you should be able to see the application.

How to setup and run:

Runs the app in the development mode Open <http://localhost:3000> to view it in your browser.

Extract the contents of the file.

```
cd <extracted-folder>
```

```
npm install
```

```
npm start
```

How to run tests

`npm test`

Build for production

Builds the app for production to the build folder. It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes. Your app is ready to be deployed!

`npm run build`

If it was a bigger project:

- Would have chosen Typescript.
- Would have used React Forms where more fields were needed to be validated.
- Using Redux/Context for API calls and data flow.