

PROJECT SPECIFICATION

FEND Capstone - Travel App

Development Environment and Architecture

| CRITERIA | MEETS SPECIFICATIONS |
|----------------------|---|
| Architecture | <p>The project should have a structure like the one shown below. All files shown must be present (Webpack may be split into multiple config files, and names may differ) and the app must successfully render a home page with clear design and functionality added when index.html is loaded in the browser. No errors should display in console.</p> <ul style="list-style-type: none">– Root:<ul style="list-style-type: none">– <code>package.json</code>– <code>readme.md</code>– <code>webpack.config.js</code>– src folder<ul style="list-style-type: none">– server folder<ul style="list-style-type: none">– <code>server.js</code> (name will vary)– client folder<ul style="list-style-type: none">– <code>index.js</code>– html/views folder<ul style="list-style-type: none">– <code>index.html</code>– js folder<ul style="list-style-type: none">– <code>app.js</code> (name will vary)– styles folder<ul style="list-style-type: none">– <code>style.scss</code> (name will vary – may be broke into partials) |
| Webpack | Webpack config should contain at least 3 scripts, express server, build and test. Additionally, dev server may be included. |
| Testing | There should be at least one test for the express server and application javascript |
| Offline capabilities | The project must have service workers installed. |

HTML & CSS

| CRITERIA | MEETS SPECIFICATIONS |
|-----------|--|
| Usability | All features are usable across modern desktop, tablet, and phone browsers. |
| | |

| CRITERIA | MEETS SPECIFICATIONS |
|----------------|---|
| Styling | Styling is set up in a logical way. All interactive elements have hover states. |
| HTML Structure | HTML structure should be indented properly with classes and ID's that make sense. |
| Visual Design | The design should clearly be different from the design used in projects 3 and 4. |

API and JS Integration

| CRITERIA | MEETS SPECIFICATIONS |
|--|--|
| Server <code>src > server > server.js</code> | Server should be a near duplication of project 3 with the exception of additional added member: value pairs. |
| index.js <code>src > client > index.js</code> | |
| app.js <code>src > client > js > app.js</code> | <ul style="list-style-type: none"> At least one function should be imported. At least one event listener should be imported. (styles referenced in html/css) |
| Extend Options / Ways to Stand Out | 1. There should be URLs and API Keys for at least 3 APIs, including Geonames, Weatherbit, and Pixabay. You can feel free to use more than 3 APIs. 2. There should be a primary object with placeholder member value pairs. 3. There should be a primary function that is exported to <code>index.js</code> . |
| Extend Options / Ways to Stand Out | At least one option from the Extend your Project/Ways to Stand Out sections have been added. Please add a Note to your reviewer which one you chose to implement, or add into your <code>README</code> . |

Documentation

| CRITERIA | MEETS SPECIFICATIONS |
|----------|---|
| README | A <code>README</code> file is included detailing the app and all dependencies. Other requirements: The Readme file should have non-default text in it that is specific to this project. It doesn't have to be thorough, but should have some basic info. Bonus points if correct markdown is used. |

| CRITERIA | MEETS SPECIFICATIONS |
|--------------|--|
| Comments | Comments are present and effectively explain longer code procedure when necessary. |
| Code Quality | Code is formatted with consistent, logical, and easy-to-read formatting as described in the Udacity JavaScript Style Guide . |

Suggestions to Make Your Project Stand Out!

At least one of these is required, but the rest are great additional ways to further customize and improve your project!

- Add end date and display length of trip.
- Pull in an image for the country from Pixabay API when the entered location brings up no results (good for obscure localities).
- Allow user to add multiple destinations on the same trip.
 - Pull in weather for additional locations.
- Allow the user to add hotel and/or flight data.
 - Multiple places to stay? Multiple flights?
- Integrate the [REST Countries API](#) to pull in data for the country being visited.
- Allow the user to remove the trip.
- Use [Local Storage](#) to save the data so that when they close, then revisit the page, their information is still there.
- Instead of just pulling a single day forecast, pull the forecast for multiple days.
- Incorporate icons into forecast.
- Allow user to Print their trip and/or export to PDF.
- Allow the user to add a todo list and/or packing list for their trip.
- Allow the user to add additional trips (this may take some heavy reworking, but is worth the challenge).
 - Automatically sort additional trips by countdown.
 - Move expired trips to bottom/have their style change so it's clear it's expired.