Zach and Alex Meet – 3-17

Integrating videos and diagrams into the app for visual learners where textual descriptions may not suffice.

Videos – Offer to download in lower resolutions to provide stronger storage.

Offline Access – Key portions of the app, such as troubleshooting guides and step-by-step instructions and diagrams are fully accessible without an internet connection. We could pre-load the most common

Alex chooses a backend technology stack which is node.js with express and MongoDB.

Alex and I choose react Native for our frontend development

We have decided to use Figma to create a MVP,

Essential MVP Features:

* The ability to send and receive messages,
* display responses from API,
* include feedback mechanism.
* Offline access
* Search functionality for past solutions
* User account management

Longer Term Upgrades

* Integration with price book
* Gamification through repair progression levels
* Allow users to attach images or videos In their messages

Zach and Alex meet 3-21

Absolutely, I'm glad we could get it working together! Let's summarize what we accomplished:

**Following day:**

Alex and I outlined the architecture for our app, which she explained to me is deciding on the structure of both the front-end and back-end and how they will communicate. We planned the main screens like the UI,

Alex taught me that managing an applications state refers to tracking how data changes over time in response to user actions or system event. She mentioned that in React Native, state management is crucial because it determines how the app behaves and renders the UI. She explains that context API is suitable for passing down data without prop drilling

We decided to use context API for some parts and Redux for others. This will help our app remain manageable as the app grows.

Alex patiently explained how to install git and how to add this initial repository into my files. She showed me that I should use the git init to initialize the file in the correct folders, and then explained git add command.

Git add is used to stage all my files for the initial commit. When I stage files, I tell git that I want to include updates to these specific files in the next commit. This is useful because it allows me to selectively commit changes.

She explained a typical process is to add all the changed files with git add to the staging area which has yet to affect the repository but marks the changes for inclusion in the next commit. Once I am happy with the changes I run git commit which takes everything that is stages and makes a new commit, which is like a snapshot of my repository. It is best practice to stage code files that I have created or updated, other project files like .json configuration files or .md documentation files and resource files that are part of the app like images, fonts, and more in the ‘assets’ folder.

Alex told me that its not typical to stage dependency folders like ‘node\_modules’ in my case, build output like compiled code or build directories, and local configuration files that are specific to my development machine and not relevant to others. I can add these untracked or ignored files using .gitignore file in the root of the repository. She let me know I can always use the **‘git status’** command to see which changes are stated and which are not, and which files are untracked by git.

Alex then explained that I should initialize a new project name plumbing prodigy and install folders like frontend and backend to stay organized. The **backend directory** is where the server code will live. I will also manage my API endpoints, data base connections and server side logic here. The **frontend** directory contains my react native code that includes my components, styles, images and other assets for the user interface portion of the app.

3-25 Meeting:

Alex and I spent the day interviewing our new candidate paul, who we are super excited about bringing on to the team to handle the technical aspects of our project. With Paul’s help we set up nodemon to assist with our development. We also added some Javascript into our server.js and plumbingAdviceRoutes.js files. It was fun to see that on the browser.

3-26 Meeting:

Zach and Alex made a big decision to switch to Python, it took the whole day to become familiar with the various python libraries and choose the libraries and tools we will use. Overall it was a success as we were able to install Django, virtual environments and more. We continue working tomorrow!