

We are a bit behind compared to our original timelines. Most of the functionality is coming together now, but we were hoping to be mostly done at this point, so we can just focus on polishing and documenting over the last week. We have accomplished a lot already but have several essential tasks still outstanding. The data models ended up being more complex than we initially thought. Below is a big-picture breakdown of where we're at with each feature/page. (Please note that we have 5 people on our team and many of the major parts are just starting to be merged in, so surprise issues could still arise over the next few days.)

- User management (login, signup, and user profile management pages): the user interface is built and mostly styled. The back-end portion is getting merged in today. It will likely need some cleanup and/or additional work to flesh out the error-handling feature.
- Recipe list (recipes page): Mostly functional and styled. It pulls the recipes from our database but still uses placeholder thumbnails until we finish implementing the image-handling processes. The image upload (back-end) feature is getting merged in today, then we just have to make sure it works with the front-end properly. We still want to add a filtering by recipe category feature.
- Recipe page: Functional and styled. May need some additional styling in the end.
- Tags: While we do have tags (string properties in the database), we ended up using recipe categories instead. (It just made more sense from a user-friendliness perspective, as we built out our wireframes.) The user can choose the recipe category as they add a new recipe, but the category filter still needs to be added to the dashboard and recipe list pages. The calorie counter feature is implemented, the values are automatically added up.
- Schedule/dashboard: Originally we were going to have two different pages to edit and view the meal plans, but ended up combining them to provide a better user experience. Most of the critical functionality is complete but it's still lacking styling. We are currently trying to implement a drag-and-drop feature with only some edge cases remaining to be dealt with. We still want to add a recipe category filter.
- Landing page: This is a fairly simple "sales/information" page and is currently being built. We are not expecting any problems here.
- We are going to make a clone of the 'Add new recipe' page (once we are done making major changes) to create the edit recipe page. It will have the same layout but will pull the values from the database based on recipe id. Then the user can make changes and save it.

From the technology stack we got approved for, we used everything but Redux. (I will send a separate email to request removal of Redux from our project plan.) We are, at this point, happy with our choices. We feel that .NET Core Web API, React, and MySQL are allowing us to solve our project problem, to allow a user to easily schedule their meals in advance, saving them time and providing peace-of-mind.

The only significant changes we made since our project plan was approved have been combining the dashboard and schedule pages, and not using Redux to manage global state.

So far, we haven't asked for help in the Homework Help Slack channel as we have been able to solve our own technical challenges. We have a team of helpful problem-solvers, and we stay in touch all day using a Discord channel we opened for just this purpose. This allows us to send messages, voice chat, split into smaller teams to discuss specific issues, and share our screens with ease. Communication has been crucial to our success. Most of our challenges - the toughest one being implementing identity - were solved by coming together and thinking/working through the problems as a team, turning it from our biggest hurdle to our biggest success.

Our most significant challenges to date have not been technical in nature. We have been struggling a bit with planning and time management, as none of us have experience working on a full-stack software project of this size, especially as part of a team. That being said, after some frustrations about a week into the project, we got back on track and in control by establishing more regular and open communications/channels and breaking tasks down into smaller pieces when claiming/assigning.