Puddlejumpers Capstone Report

**Table of Contents**

[Final Project Report](#_5olnvqkrjmuz)

[Did you complete all in-scope tasks?](#_u47mzy2krmd1)

[Did you complete any extra tasks?](#_jgiq6fejoblx)

[Drag-and-drop feature (dashboard)](#_yegpry35uoa2)

[Edit mode (dashboard)](#_bm2lmbfoh752)

[Identity (more complex than our original plan for user management)](#_7emtkadu7log)

[Satisfaction Assessment](#_rovt44ppntiq)

[Does your app solve your problem?](#_32b3jmt0gwto)

[Have you or someone else started using the app? What do they think of it so far?](#_wyzc17mco2b2)

[Lessons Learned](#_pr3fyf839cme)

[Spend more time planning](#_2n60sez4p13g)

[Communicate sooner and more](#_uyukt22dtja1)

[Identity](#_19zqtsgkunte)

[Sometimes it’s okay to eject from Create-React-App / Implementing Tailwind CSS](#_hx7wiyk4ibr2)

[Don't put your seed data inside Entity, keep it separate.](#_825qdz50bkyl)

# 

# Final Project Report

## Did you complete all in-scope tasks?

Yes, we completed all the in-scope tasks.

**In-scope features**

- Adding a recipe

- Viewing a recipe

- Editing/updating a recipe

- Deleting a recipe

- Viewing meal plan (schedule)

- Making/editing meal plan (schedule)

- Add a user profile

- Edit a user profile (update name/password)

**Out-of-scope features**

- Password email flow - to send password reset link to user's email

- Delete a user profile

## Did you complete any extra tasks?

Yes, we completed some extra tasks:

### Drag-and-drop feature (dashboard)

### Edit mode (dashboard)

### Identity (more complex than our original plan for user management)

# Satisfaction Assessment

## Does your app solve your problem?

Yes, Yummy solves the problem of saving time and headache for a busy user by allowing them to plan their meals in advance. When designing Yummy, we focused on keeping it simple and user-friendly.

## Have you or someone else started using the app? What do they think of it so far?

No, we did not have the time to have the application tested by others.

# Lessons Learned

If you had to build another full-stack CRUD app what lessons would you apply that you have learned from this project?

## Spend more time planning

One lesson we definitely agree on having learned is that if we were to do this again, we would divide the pages/features into much smaller tasks before beginning to claim/assign them. Not having worked on such a comprehensive software development project before, we took the initial planning in a stride and were so eager to get started with the actual building, that we agreed that planning in too much detail will just end up being a waste of our time. We did create an ERD and wireframes, but did not take the time to research what technologies/libraries we were going to use to build each of the features. This led to a lot of redundant work in the end as we figured out things as we went. That being said, at least some of the extra work was not caused by lack of detailed planning but rather by lack of experience. We didn’t know what we didn’t know, as the saying goes.

## Communicate sooner and more

Another thing we learned was that communication is even more essential on a 3-4 week project than it is on a 1 week project (our milestone 2). We started the project with meetings but also spent a lot of time individually working on our own parts. It was about a week and a half into it that we realized that we needed to communicate more/better. We set up a Discord channel with places to send messages and files, and three different meeting rooms where we could split into smaller groups and have audio discussions with screen sharing. Having a place where we could just jump into or even drop by, unplanned, did wonders to our productivity! If we were to do this project all over again, we would set up the Discord channel on day 1!

## Identity

We struggled implementing Identity into our application as Identity starts off using the MVC structure in C#. Identity prepopulates user registration and authentication pages in razor view which could not be used in our react project. We decided to implement Identity as a base into our project for the data models, but we would create a custom controller file to handle the data models that Identity creates to handle our authentication. This proved to be a great lesson in implementing Identity as a base but scaffolding out to handle custom API endpoints which our project would communicate with.

## Sometimes it’s okay to eject from Create-React-App / Implementing Tailwind CSS

## Don't put your seed data inside Entity, keep it separate.