

Agenda

- Progress Report
 - Isabel
 - Yummly recipe mining
 - Code to populate initial database with recipes
 - Whole pipeline for retrieving recipes
 - Lily
 - UI and backend for Pantry (get/add/delete ingredients)
 - Bootstrap
 - Maryann
 - UI and backend for Search
 - Database relations
 - Nadia
 - Set up Mongoose
 - Users and user authentication
 - Home page
 - Database relations
- Questions
 - For adding ingredients, should it be a put request or can it be a post request?
 - How much middleware?
 - Nested get requests?
 - How to have new pages navigate to different urls instead of just replacing stuff on the same page (i.e. recipe/:id)
- Goals for week
 - Next group work session availability
 - Isabel: ratings, hook
 - Lily: ingredient amounts and active/inactive ingredients
 - Maryann: searching by ingredient amounts and recipe scaling
 - Nadia: implement cookbook
 - MVP due 11/23
 - Reflection due 11/24
 - Halfway point 11/29

Minutes

- Went through progress for each team member
- Asked about redirection to /recipe/:id with actual different pages. Lily mentioned she could help with this. Harihar brought up two approaches: a client side routing library or just using ejs to serve files from the server
- Asked about put/post to pantry: use put requests for pantry updating
- Search get request: repopulates page via loadHomePage method ok? Yes it's ok
- Middleware: check for ownership!

- Gulp: write a script that watches source files, if they change then it auto recompiles them. nodemon automate server. Consider setting currentUser to my User object for development
- Add middleware and cookie stuff to save user logins
- Disable caching
- Yummly does not provide instructions in API - it's ok just to use URLs
 - consider taking a screencap of the page