

- DB design
- Set deadlines
 - Code: due on October 7th
 - Worksheet: end of day October 10th
- Go over AWS account and how the console works
 - I am going to make the account, but please be conscious that it's connected to my credit card.
- Setup meeting time with the TA.
 - I don't think they actually read my email since they only gave me a single time (5:30 on Wednesday), plus they didn't give me their github login information.
 - Lol yeah i dont doubt it
- Amplify (deployments)
 - Don't touch the api or auth folders under amplify.
 - For the lambdas we write, don't touch the format.
 - We can use Postman to connect the api endpoints from the backend.
 - We can add as many controllers as we need for the api connections and other things.
 - There are a bunch of files being automatically generated by amplify.
 - We'll need to look at different Amplify command-line arguments for pushing our code and updating the environment.
- Amplify (console):
 - We won't be touching this much on the console's end.
 - Don't push to the dev or main environment since nothing would get touched by the tests. We'll be pushing to these branches using Jenkins!!!
 - SO ONLY CONNECT TO THE TEST ENVIRONMENT.
 - If we're doing something with the TEST environment, we'll need to communicate!! #touching-test
- We use us-east-2 only.
 - We need to install the AWS CLI locally, and make sure that it's using this region, or else it'll deploy to another region, which we don't need.
- Main .ts file is setup
 - This is for the Frontend people. We use the index.ts file for most of the main structure of the application.
 - We can use models to populate this.

- We would pull libraries with new models in the api-models folder.
- AWS setup:
 - Use the credentials file that is shared in discord.
 - When first logging in with that IAM user it'll ask you to change your password and then setup a secret key for pushing changes. => Create your access key and secret key and download them
- API Gateway:
 - We can test the API endpoints through the console with this service.
 - This is only for manual testing, but we will be using Postman to automatically run and send endpoints for our own testing!
- Amazon Cognito (service):
 - This is to create users and where we store our users.
 - We'll need to look into authenticators, which can be done for the login page.
- AWS Lambda (console):
 - Lambda is running code that runs 24/7 on the cloud. When the request comes in, it runs the code automatically.
 - There are a lot of different plugins and imports that can connect to Lambda.
 - We can use very small Lambdas with many different API calls to make it run quicker (load balancing!!).
 - Amplify can automatically generate templates for Lambdas with the API gateways. It will handle creating the file and the different configurations it would need.
 - Here we will need to modify the config part for the Lambda which changes which services it can connect to.
- AWS Lambda (development):
 - To run the Lambdas, it is best to use Rider (best UI).
 - LambdaEntryPoint.cs:
 - It is the starting program that run startup.cs
 - launchSettings.json:
 - This is how we are actually able to run all of the tests/lambdas locally. It's a part of the git repository and if it doesn't work, let Qiwen know!
 - Make sure you have the .NET path set correctly within the environment of the Rider IDE.
 - Dotnet 6 !!!!!!!

- DynamoDB:
 - Amplify prefers this.
 - `aws configure —profile`
- Action Items:
 - Pull in the project from GitHub.
 - Pull in AWS CLI onto your machine.
 - Setup AWS Amplify on your local machine/ide.
 - Log into the AWS Console with your credentials.
 - Create an account for Rider.
 - Cynthia: make the spreadsheet bigger for dev tasks since we have a lot to do with the distributed aspect of this.
 - Read through these minutes again for the notes Tara took from Qiwen's explanation.
 - Qiwen please update these :D
 - Get dev tasks done for FRIDAY.
 - Meeting room booked for Friday - Kelly
- Linux backend for the EC2 instance.
 - This is a server that we'll use to run the application.