Grading Requirements  
This project is graded based on the following criteria:Technical Acceptance Criteria: 25%

* Satisfies the following code requirements:
* Application uses a Node.js and Express.js back end and uses both GET and POST routes for retrieving and adding new data.
* Application has a folder structure that meets the MVC paradigm and uses Handlebars.js as the template engine.
* Application is backed by a MySQL database with a Sequelize ORM and protects API keys and sensitive information with environment variables.
* Application includes user authentication (express-session and cookies).
* Application uses at least one new library, package, or technology not covered in class.

Concept 10%

* Application should be a unique and novel idea.
* Your group should clearly and concisely articulate your project idea.

Deployment: 20%

* Application deployed at live URL on Heroku and loads with no errors.
* Application GitHub URL submitted.

Repository Quality: 10%

* Repository has a unique name.
* Repository follows best practices for file structure and naming conventions.
* Repository follows best practices for class/id-naming conventions, indentation, quality comments, etc.
* Repository contains multiple descriptive commit messages.
* Repository contains quality README file with description, screenshot, and link to deployed application.

Application Quality: 15%

* Application user experience is intuitive and easy to navigate.
* Application user interface style is clean and polished.
* Application is responsive.

Presentation 10%

* Your group should present using Powerpoint or a similar presentation software.
* Every group member should speak during the presentation.
* Your presentation should follow the [Project Presentation Template](https://docs.google.com/presentation/d/1_u8TKy5zW5UlrVQVnyDEZ0unGI2tjQPDEpA0FNuBKAw/edit).

Collaboration 10%

* There are no major disparities in the number of GitHub contributions between group members.

Requirements:

* Use Node.js and Express.js to create a RESTful API.
* Use Handlebars.js as the template engine.
* Use MySQL and the Sequelize ORM for the database.
* Have both GET and POST routes for retrieving and adding new data.
* Use at least one new library, package, or technology that we haven’t discussed.
* Have a folder structure that meets the MVC paradigm.
* Include authentication (express-session and cookies).
* Protect API keys and sensitive information with environment variables.
* Be deployed using Heroku (with data).
* Have a polished UI.
* Be responsive.
* Be interactive (i.e., accept and respond to user input).
* Meet good-quality coding standards (file structure, naming conventions, follows best practices for class/id naming conventions, indentation, quality comments, etc.).
* Have a professional README (with unique name, description, technologies used, screenshot, and link to deployed application).

[9:48](https://uciirvfsfpt11-pbl5234.slack.com/archives/C01ERCZMJTV/p1615870085001300)

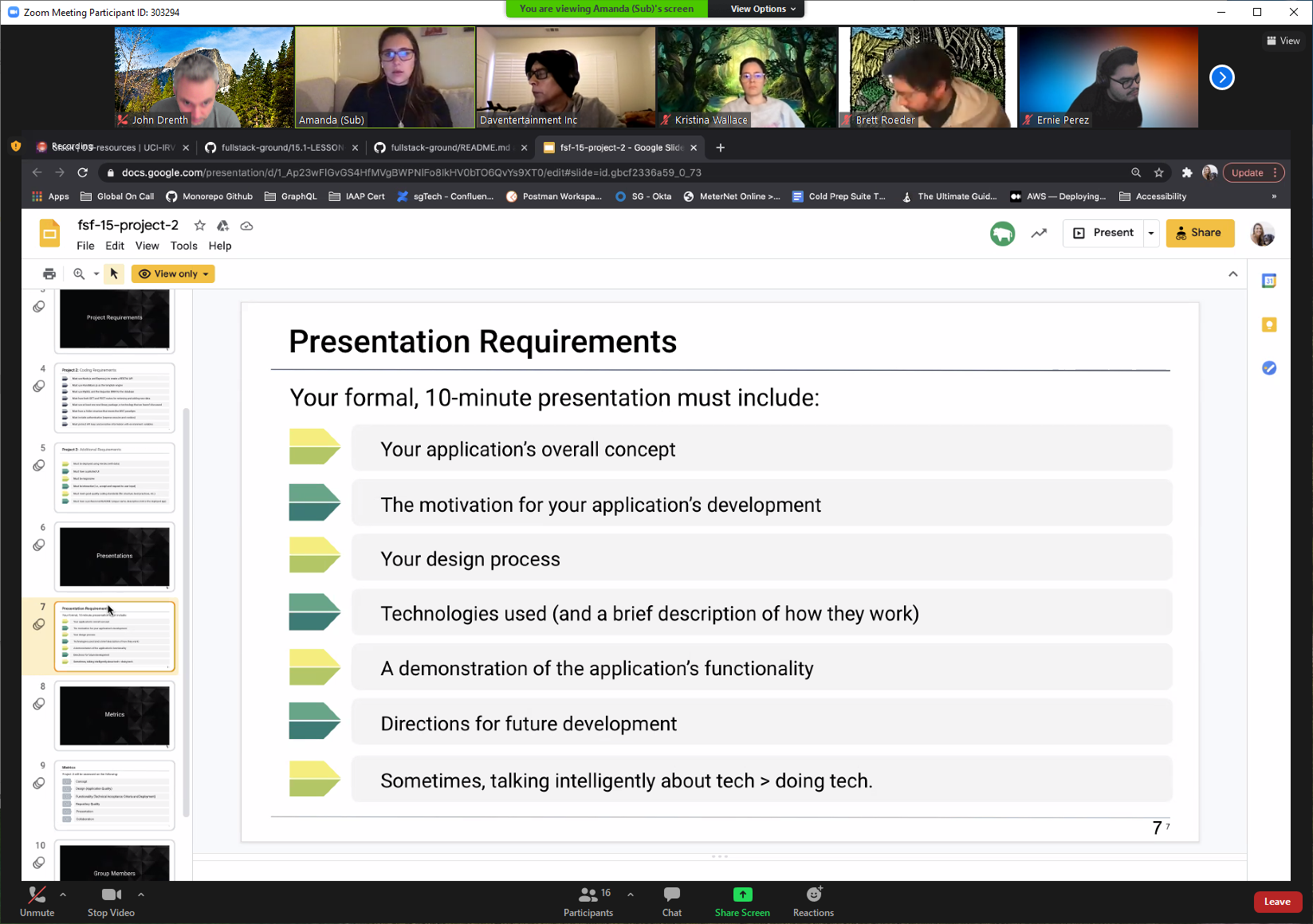
Presentation Requirements  
Use this [project presentation template](https://docs.google.com/presentation/d/1_u8TKy5zW5UlrVQVnyDEZ0unGI2tjQPDEpA0FNuBKAw/edit?usp=sharing) to address the following:

* Elevator pitch: a one minute description of your application
* Concept: What is your user story? What was your motivation for development?
* Process: What were the technologies used? How were tasks and roles broken down and assigned? What challenges did you encounter? What were your successes?
* Demo: Show your stuff!
* Directions for Future Development
* Links to the deployed application and the GitHub repository. Use the [Guide to Deploy with Heroku and MySQL](https://coding-boot-camp.github.io/full-stack/heroku/deploy-with-heroku-and-mysql) on The Full-Stack Blog if you need a reminder on how to deploy to Heroku.

**coding-boot-camp.github.io**

[**Deploy with Heroku and MySQL**](https://coding-boot-camp.github.io/full-stack/heroku/deploy-with-heroku-and-mysql)

In this guide, we'll walk through the steps we need to take to deploy an application to Heroku with a MySQL database.



Your application’s overall concept

The motivation for your application’s development

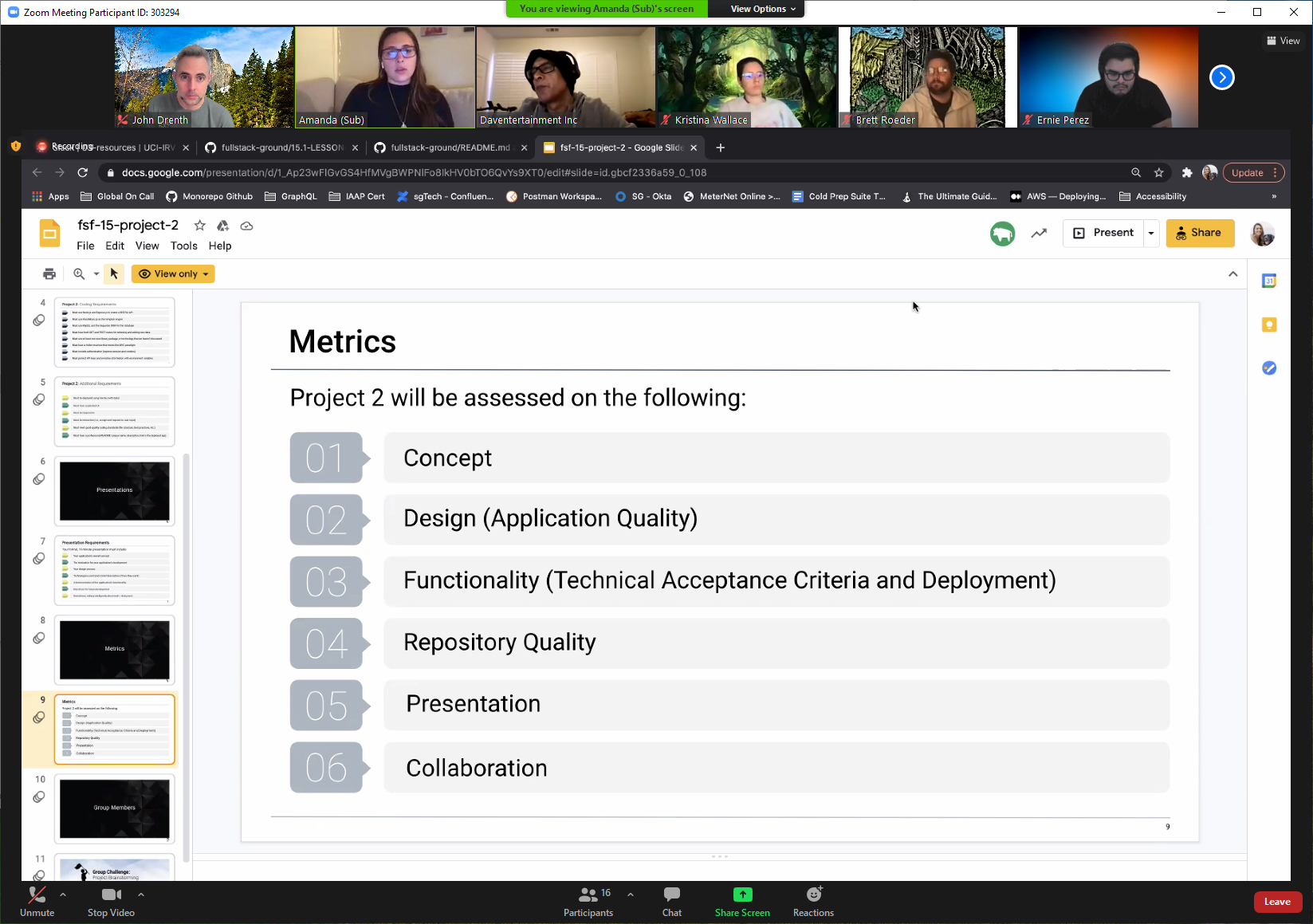
Your design process

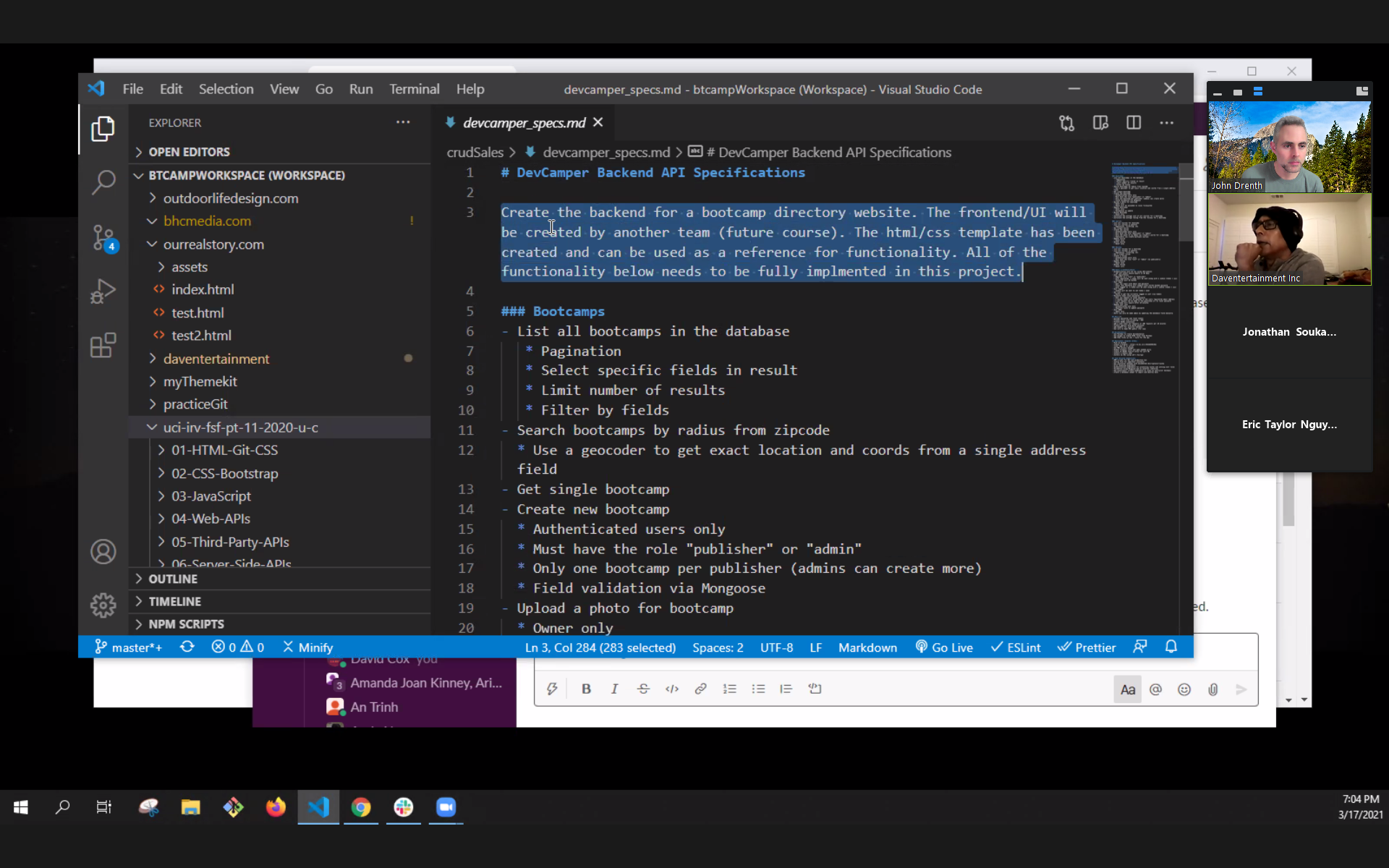
Technologies used (and a brief description of how they work)

A demonstration of the application’s functionality

Directions for future development

Sometimes, talking intelligently about tech > doing tech





Best Practices:

* + Use Trello
  + App Idea (manageable in 2 weeks but still fun & interesting, what’s going to make it unique, what does it provide that others don’t have)
  + Break down the work into all the steps that need to happen
    - Create user story for each step…
    - As a user I want to register for the app
    - As a user I need to be able to login
    - Need Authentication
    - Need GET
    - Need POST
    - Upload photo?
  + If someone is artistic they could make a wireframe

APP IDEAS:

1) Build an online post it app, ppl can interact, throw ideas out, etc.

2) ~~Pet Life Tracker~~ **~~Pet Gazette~~**

**~~3 lucky Paw~~**

**~~Paw Pal~~**

**~~Paw-ssible~~**

**~~magic paw~~**

**~~paw Protector~~**

**~~happy Paw~~**

**3 paw-sperity**

- User Registration

- User Login

- Add Pet

- Pet Categories (Dog, Cat, Rabbit, Bird, Rodent, Dinosaur)

- Vet, doctors at office

- Vet Visit (last one, next one, activity performed..)

* + Reason (vaccination, general checkup, xray)

- Grooming, types (haircut, nail trim, pet bath)

- Pet Food

- Medications

- Types (digestive, antibiotic, painkiller, etc.)

- Course (start date, end date, frequency, dosage) end date optional

-Pet Types

Food, treat,

3) Doctors office app