# **Capstone Project:** Food2Go

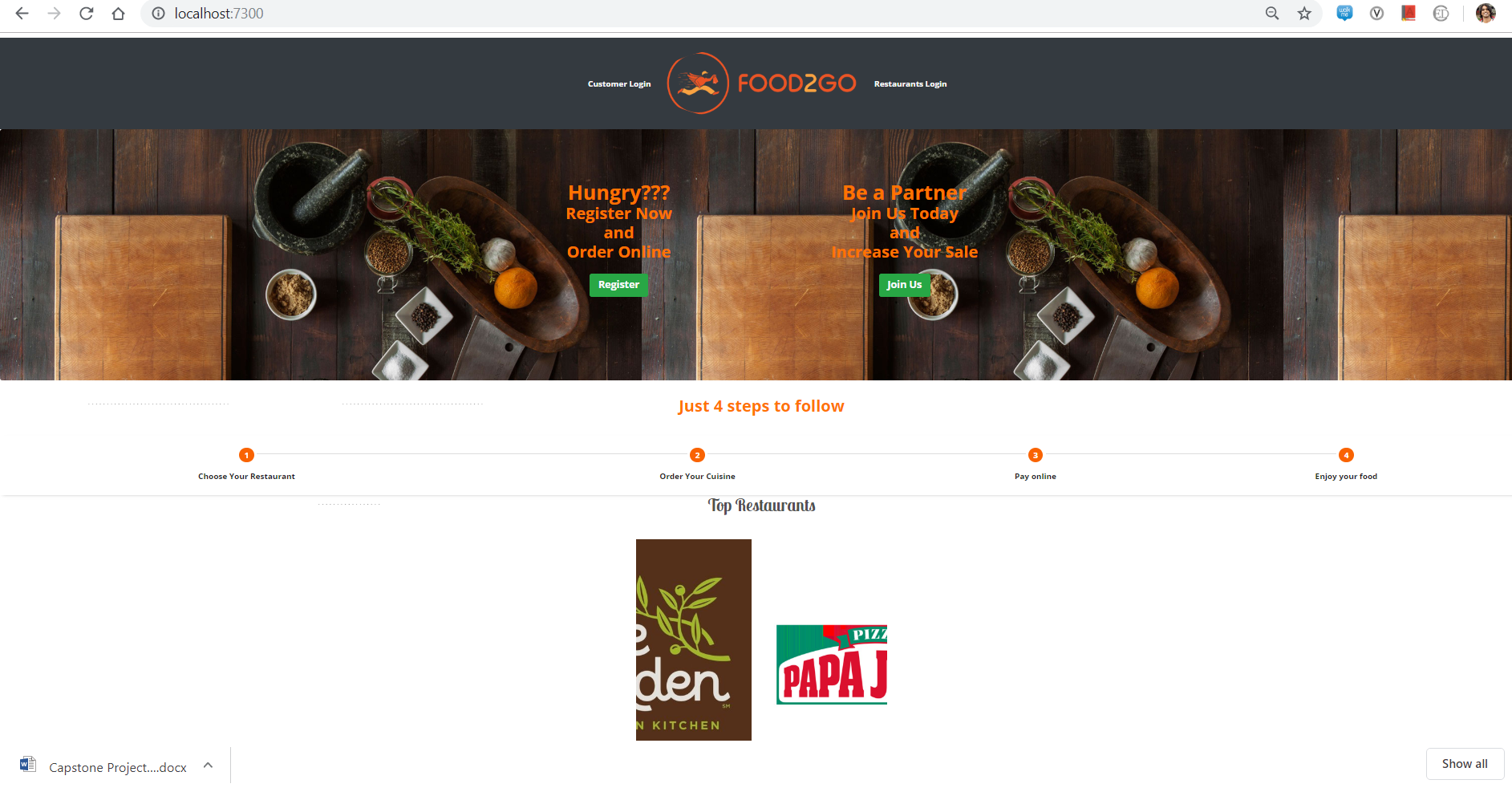
# **Student Name:** Imtiaz Ahmed Kapatrala

A url to the running web application

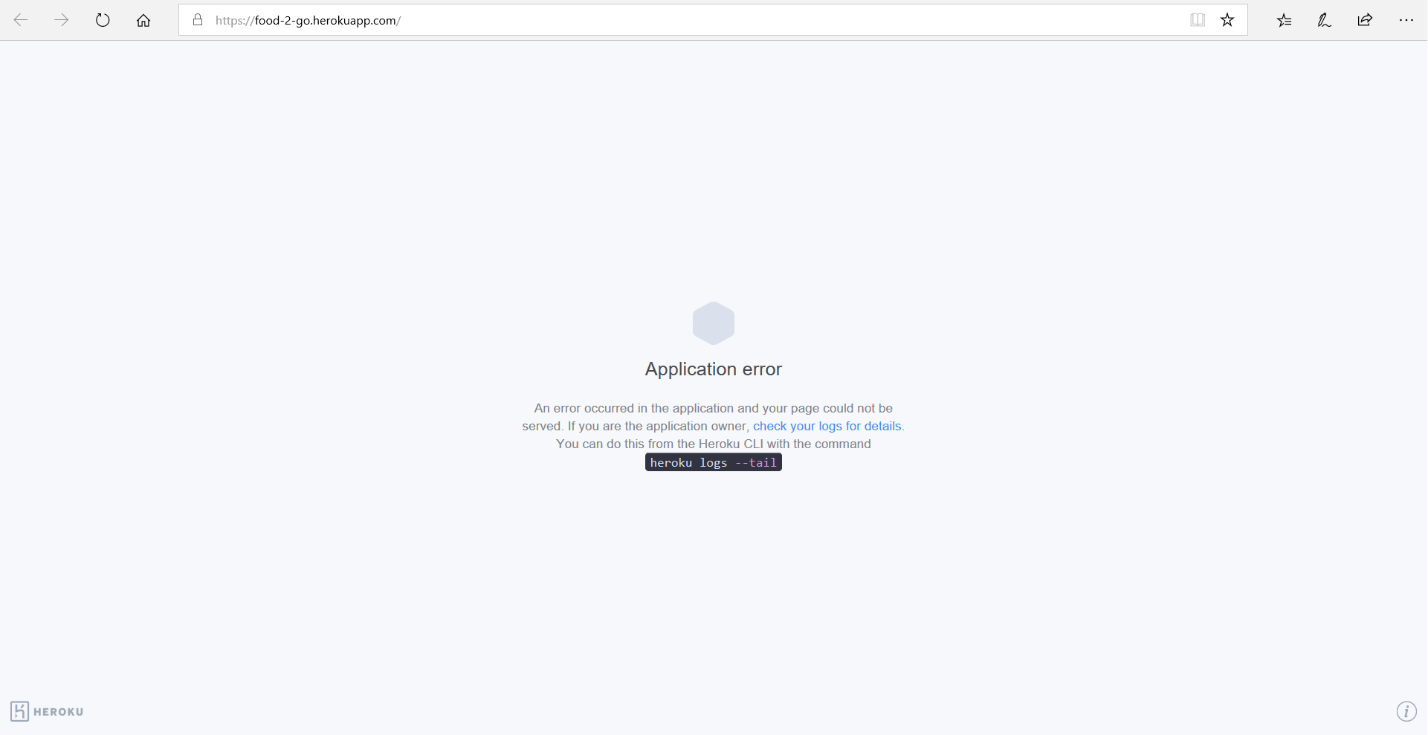
I tried deploying the app both on AWS and Heroku. But I was unsuccessful.

However, If you pull the code from the below GitHub repo, and run the code, the application will be up and running.

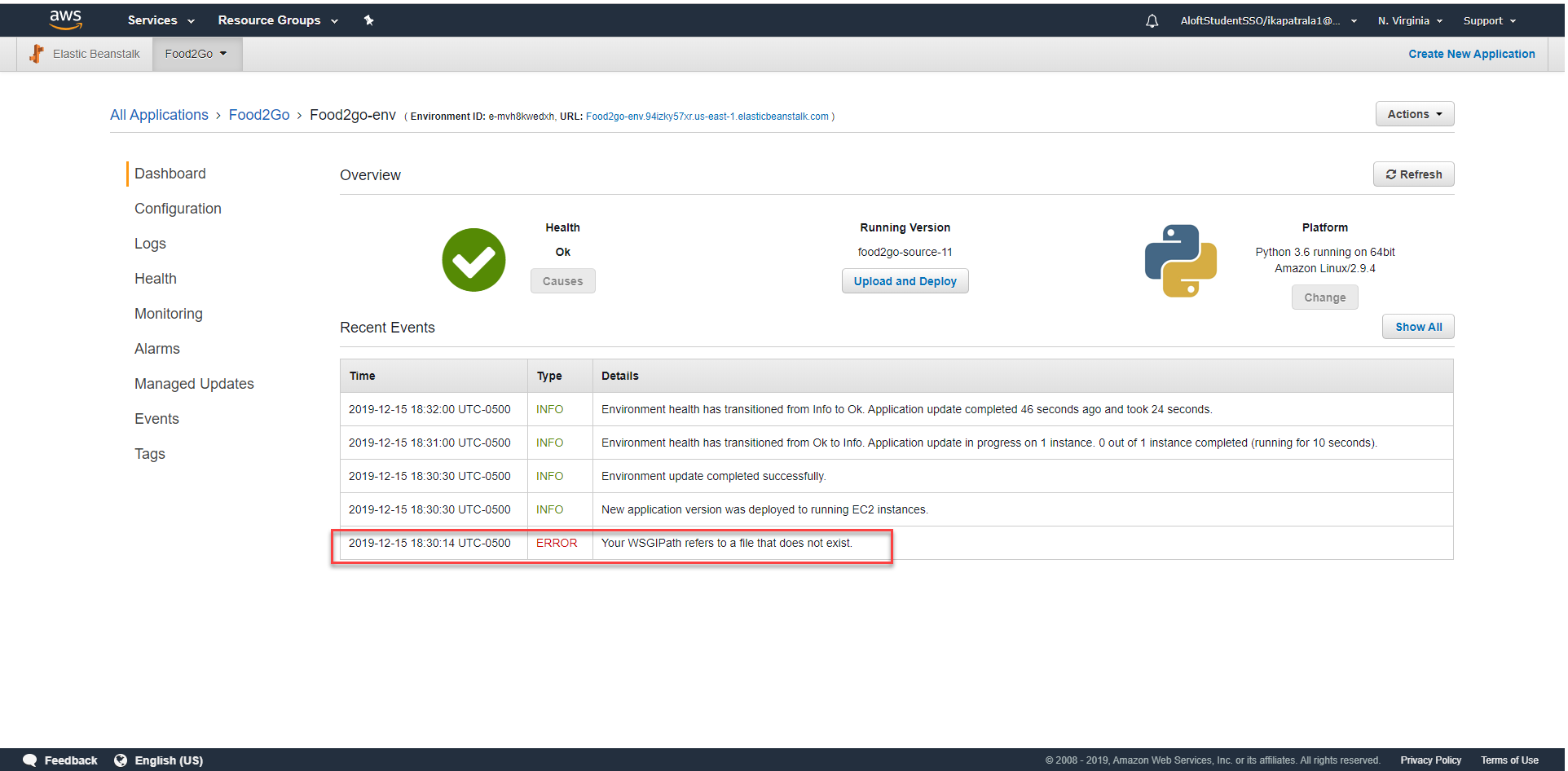
<https://github.com/ikapatrala1/Food2Go-App/tree/master/Food2Go>



**Heroku:**



**AWS:**



The VPC id for your running application servers

* + If you have an agreement with the instructor to use a deployment different than AWS, then please include any necessary identifiers for the location of your servers

As I mentioned above, I was unable to deploy to code to AWS.

If kept outside of GitHub Projects (e.g., you used Visual Studio Team Services or Trello), then a link to your final Backlog/Kanban that should be up to date with completed, and incomplete, stories and tasks

<https://dev.azure.com/ikapatrala1/Food2Go%20App/_boards/board/t/Food2Go%20App%20Team/Issues>

"What is the most difficult **design** challenge you faced on this project, and how did you handle it?"

Designing the data models was a somewhat challenging when I first started building the app. I leveraged the model feature of the Django framework, which saved a lot of time. Django comes with robust database management features. Generally speaking, these features allowed me to do two things; 1) design database schemas; i.e., tables, relationships, etc, 2) manage the data in a database; i.e., write, read, update, delete, query, etc.

"What is the most difficult **technical** challenge you faced on this project, and how did you handle it?"

While implementing the authentication for the Food2Go app I had some technical hurdle. I used the Django authentication which handles both authentication and authorization. Here, authentication verifies a user i.e. it is who they claim to be, and authorization determines what an authenticated user can do. The auth system consists of:

* Users
* Permissions: Binary (yes/no) flags designating whether a user may perform a certain task
* Groups: A generic way of applying labels and permissions to more than one user
* A configurable password hashing system
* Forms and view tools for logging in users, or restricting content
* A pluggable backend system

"If there is one thing you could change about the project, how would you change it and why?"

Before deploying any Django web app in the real world, we need to make sure the project is production-ready. The best practice is to start implementing them earlier. It saves a lot of time and headache. I could have started the deployment early to avoid deployment issues. Going forward, I will make sure by at least deploying a sample application into AWS or Heroku.

"After working on this project, and now submitting the final product, do you feel like you can develop software? Why or why not?"

My experience with this project has been extremely positive. The Capstone project gave me an insight into the end-to-end development of any project. It is an excellent opportunity to showcase my knowledge that I have learned as part of this course.

Before this project, I was not aware of how to propose an idea and create a successful proposal. I didn’t know about pitching the project, documenting the MVP, and project planning. However, after completing the project, I can say I am no longer ignorant nor uninformed. Because of this project, I feel like I can contribute more to my current project in my organization. I can make more informed decisions and share what I have learned with others in the team.

Overall, I am pretty much sure that I can handle any software project with ease by having these skills.