

# CIS 3368 Fall 2021 Final Project

## Deciding Dinner

For your final project you will be building a web application following a standard 3 tier architecture:

### **UI – REST API Services – Database**

This application will help users to decide on a place to have dinner at.

Initially, everyone in the dinner party should create a profile, with at minimum a field for **id**, **first name** and **last name**. The id assignment should happen automatically. All users are stored in the user table.

Next, every user should be able to enter 5-10 restaurants, with 5 being the minimum, and 10 being the maximum allowed entries per user. These restaurants need to be stored in their own restaurant table.

There should be edit pages for properties on the restaurants and users (e.g. to change a user's name).

Finally, there needs to be a page for the decision making: here one should be able to select who is going to dinner among all available users. Then among all the selected users' restaurants, an algorithm should select one at random and present it on the screen as the chosen restaurant for this night's dinner.

### *Requirements*

This project will be developed in **2 Sprints**. The first sprint focuses on the Python backend development, all the logic/functionality, and the deployment of the REST API services with Flask. The second sprint focuses on the UI development in JavaScript using the EJS. Both **code submissions** will happen into the same repository on GitHub Classroom. Do not forget to comment and document your references.

Each Sprint for the project is due at midnight of the listed due date. No late submission will be accepted. Part of the submission for Sprint 2 will be a 5-minute long **presentation video**, giving an overview over the functionality of your application features, and explaining the technology, frameworks, etc. that you have used.

### ***Details for Sprint 1 - Deadline October 16th, 2021***

For Sprint 1 you will have to design and implement all the services you will need so that your app can interact with your database (**ALL CRUD operations for the two tables**), as well as an API that makes the random restaurant selection. These REST Services need to be implemented using the Python Flask Framework. You will be testing your code without the need to have the pages implemented that make

up the web-based user interface for the app.

You will only have to submit the link to your github repo in Blackboard.

## ***Details for Sprint 2 TBA - Deadline December 4<sup>th</sup> 2021***

### ***Other Requirements***

- Make sure the username and password for the database you setup are credentials you're willing to share. Do not use personal passwords for this homework, which you might be using anywhere else.
- Make sure your project compiles and runs. If the project doesn't run, you will forfeit points.

### Using Github Classroom to setup the remote repo for this assignment

The link to accept the assignment and create the private GitHub repo for the Python and JavaScript code will be posted in Blackboard. Accept the link and an empty GitHub repository will be created for you. Please push your commits on a regular basis.

### What to Turn in

Commit your source code (properly commented) to your private repository in the GitHub classroom. Do not zip the files together. Your GitHub repository should show multiple meaningful commits illustrating how you worked through the problem.

There will be 2 assignments listed in Blackboard - one for each Sprint.

For Sprint 1 submit the link to your GitHub repository containing your project via Blackboard.

For Sprint 2 submit the link to your presentation. Video **MUST** be uploaded online and accessible for everyone (verify before submitting).

### Group submissions

You can either work alone, or with 1 partner as a team of two. If you form a team of 2, you **MUST** submit a project document at least 24h prior to presentation via Blackboard. This document must outline in detail which part of the project was worked on by which team member, and how the work

was distributed. The presentation video does not have to be narrated/showcased by both team members. However, both team members must be aware of each others code. Both team members will receive the same grade for the project.

### Extra Credit

You can earn extra credit if you are able to deploy the Python Flask API and/or the web front-end into the cloud. Do not attempt this until you have finished the other parts of the project. There are many options of cloud deployment and you are free to choose one that fits the particulars of this project (Python Flask + Node based front-end).

### Basic Grading rubric for the project

Item	Points in Sprint 1	Points in Sprint 2
Remote DB Tables setup	10	
Code for API	10	
Comments explaining API code	10	
Code for interaction with remote DB	10	
Comments explaining interaction with DB	10	
Code for UI (3 Pages)		20
Comments explaining UI code (3 Pages)		20
Video presentation		10
<b>Total</b>	<b>50</b>	<b>50</b>
<i>Deduction 50% if code doesn't compile</i>		(up to -50)
<i>Deduction for not submitting link to Github</i>		-5 points
<i>Deduction for not submitting link for video presentation</i>		-5 points
<i>Deduction for not having Multiple meaningful commits showing progress</i>		-5 points -5 points
		10 points
<i>Extra credit for Cloud deployment</i>		10 points 10 points