# COMPSCI 326

**Project Milestone 3** due 5/7/22

## Overview

This project will give your team an opportunity to present an innovative web application that your team will be developing over the latter half of this semester. **In this culminating milestone, you will be implementing most of the logic and corresponding database interactions for the back-end of your project.** As before, we expect your thoughts and ideas for your project to change during the course of the semester leading to changes in the data your application will support and the interface your users will expect. **This document describes the requirements for the final submissions:**

1. **Milestone 3 (back-end implementation)**
2. **The Final Report.** Please read the Final Project Specification to understand the requirements for the final submission.

## Milestone 3

You are required to make any necessary updates to **docs/setup.md** with all the steps required to build your project. Remember to include all the source files including all JavaScript in your repo.

## Part 1: Database implementation

Your team will need to implement all functions (preferably in a separate file as we did in the in-class exercise) that your application uses to perform CRUD operations on the database.

1. First, you will need to create a database to store all the data for your application. Plan how you are storing the data and how you will reference the functions.
2. Next, once you have decided on and created the structure for your database, you will write the functions to perform Create, Read, Update and Delete operations on the Database.

You need to include an example document **for all tables/collections** in your Database to show the structure of your database. Include the Datatype and a brief description for all fields about what they are in your project. For example, our counter application for the in-class exercise would be documented as follows if you were using MongoDB:

```
counter document
{
     _id: <ObjectId1>,
     countername: String,  // The name of the counter
```

```
        countervalue: Integer,  // The value of the counter
}
```

If you are using a relational database, **show the columns, data type and description for all your tables**. For example, our counter application for the in-class exercise would be documented as follows:

```
counter table
| Column       | Data Type | Description              |
|--------------|-----------|--------------------------|
| countername  | String    | The name of the counter  |
| countervalue | integer   | The value of the counter |
```

**Important** note about using a database in production. **You must keep your secret keys to access the database (URIs and passwords) SECRET!** This requirement means you *cannot* push them to GitHub and cannot hard-code them as strings in your code. Below is the alternative method that you *need* to use in order to keep your application secure.

1. Create a file called **`.env`** in your main folder and put your passwords and URIs in this file. Use the dotenv package to load these values in.
2. Add **`.env`** to your **`.gitignore`**. If you need to share secrets with your teammates, use Slack or some other secure method.
3. You can also create environment variables manually on Heroku, which you can do by following these instructions: https://devcenter.heroku.com/articles/config-vars

You will also be creating a Markdown file called **milestone3.md.** in the **docs** folder of your github repository. It should contain:

1. **Documentation** describing your database, as spelled out above.
2. A **breakdown of the division of labor** for each team member **for this milestone** — that is, saying who did what. Remember that *everyone is expected to contribute roughly equally* to each phase of the project. We expect to see similar numbers and kinds of GitHub commits by each student. These must match the reported division of labor; any mismatches need to be explained. We expect at this point that everyone's GitHub client is properly configured.

Commit this document to your github repository by the due date and then create a "release" for Milestone 3 tagged **`milestone-3`**. There is no need to "submit" anything. A reminder: for instructions on how to create a release, see:

- https://help.github.com/en/github/administering-a-repository/managing-releases-in-a-repository#creating-a-release

## Part 2: Back-end Functionality

Implement all necessary logic of your server to connect up your data model with the user interface so that everything is working.

## Part 3: Deployment

Include the link to the hosted application in your **docs/final.md**.

For the most part, it should be as simple as linking your existing GitHub repository to the Heroku Auto-Deploy feature as described here: GitHub Integration (Heroku GitHub Deploys).

If you need additional instructions or help, you can refer to the Documentation from Heroku for more details.

## Final Write up and Video

You will need to create a markdown file **docs/final.md** and a **video demo** as submission for the final report, as well as vet your repository.

### Code Cleanup

Given this is an expansive public project, it is important that it reflect a high standard of development. As such we would like to provide your team the opportunity to scrub the repository of any bugs you have since encountered, as well as be faithful in your adoption of best practices, including the following.

- Crisp and clear documentation (from previous milestones), without spelling or grammatical errata.
- Linting of JS.
- Purging of extraneous dependencies and files.
- Removal of commented-out or unused code.
- Reasonable naming conventions for files, methods, functions, and variables.
- Separation of JS from HTML and CSS from HTML.
- Validation of HTML.

If you've been faithful in your fulfillment of the previous milestones, in addition to following course guidelines, these items *should* already be satisfied. Think of this as a last "proofread" before you "ship" your application.

### Markdown Submission

You are required to make any necessary updates to **docs/setup.md** with all the steps required to build your project. Remember to include all the source files including all JavaScript in your repo.

Create a new file **docs/final.md** file in your github repository. In this document your team must include the following document:

1. **Title**: The title should be your team name.
2. **Subtitle**: The subtitle should be your application name (if different from team name).
3. **Semester**: The semester (e.g., Spring 2022)
4. **Overview**: A brief overview of your application. This will be based on what you are submitting as your final web application artifact. You should also mention why your application is innovative.
5. **Team Members**: A list of your team members, with names and GitHub aliases.
6. **User Interface**: A final up-to-date list/table describing your application's user interface. This should include the name of the UI view and its purpose. You should include a screenshot of each of your UI views.
7. **APIs**: A final up-to-date list/table describing your application's API
8. **Database**: A final up-to-date representation of your database including a brief description of each of the entities in your data model and their relationships if any.
9. **URL Routes/Mappings**: A final up-to-date table of all the URL routes that your application supports and a short description of what those routes are used for. You should also indicate any authentication and permissions on those routes.
10. **Authentication/Authorization**: A final up-to-date description of how users are authenticated and any permissions for specific users (if any) that you used in your application. You should mention how they relate to which UI views are accessible.
11. **Division of Labor**: A **breakdown of the division of labor** for each team member — that is, saying who did what, **for the entire project**. Remember that *everyone is expected to contribute roughly equally* to each phase of the project. We expect to see similar numbers and kinds of GitHub commits by each student.
12. **Conclusion**: A conclusion describing your team's experience in working on this project. This should include what you learned through the design and implementation process, the difficulties you encountered, what your team would have liked to know before starting the project that would have helped you later, and any other technical hurdles that your team encountered.

Update your github repository by the due date and then create a "release" for Final tagged `final`. There is no need to "submit" anything. A reminder: for instructions on how to create a release, see:

- https://help.github.com/en/github/administering-a-repository/managing-releases-in-a-repository#creating-a-release

### Video Demo

You are required to record a video demo of your project as part of the final submission. The video demo should be about 5 minutes long (we will stop watching after 5 minutes). You should include at least the following:

- Project name and team name
- Introduction **FROM ALL** team members (should your situation allow, it'd be nice to see each person's face)
- Description of the application including its novelty
- Targeted users (who would use your application)
- Quick important feature walkthrough / demo
- Highlight of specific aspects of your application that your team finds most interesting (e.g., UI, back-end algorithms)
- A list of what your team found easy and what your team found hard
- Future work - what would you do if you continued to work on this application

Here's a link to previous semester projects: Web Programming (COMPSCI 326) Spring 2020, Web Programming (COMPSCI 326) Fall 2020, Web Programming (COMPSCI 326) Fall 2021

You should first accept this invite to be a "collaborator" for COMPSCI 326 Spring 2022. https://www.youtube.com/playlist?list=PLr94vpzaMmB_PYPBsHuy7ahXj2oSWwVZI&jct=3SZI_GqBFwJBoHVNvtjnQP6zBHcltA

Next, upload your video to YouTube and then add it to the COMPSCI 326 Spring 2022 playlist with this link: Web Programming (COMPSCI 326) Spring 2022

After you click the blue "continue" button at the link above, one team member should upload the video to their personal channel and then add it to the playlist *from the video's upload page* (not from the playlist page).

More details on how to add your video here: https://support.google.com/youtube/answer/6109639

As before, you should use the COMPSCI 326 Slack for discussions while you are working on your project.