

# Project: (Deadline: 12/04)

## 1. Introduction

For the final project, you will be creating a web application using the key technologies we have practiced throughout the semester. Instead of giving you a specific web application to replicate, you are given the freedom to choose what you would like to make.

Your web application needs to have three components: frontend, web server, and database server. We are giving you some restrictions on what you can make, such as a list of technical and style requirements. If you do not have any ideas for a web application you would like to make, we have also provided a Freeway project that you may use. If you choose to create the Freeway project, it will be graded in the same way the other final projects are graded.

Unlike the homework assignment, you may work on the final project in groups of 1-2 students.

Unlike the homework assignments, the final project **cannot be turned in late**.

Key project dates:

- Proposal due dates: 10/24
- Proposal presentation: 10/29
- Final project presentation: 12/03 or 12/05
- Final project due dates: 12/06

## 2. Project requirements

The final project is mostly open-ended, with some specific requirements as outlined in this section.

Overall, you can receive full credit on the final project if:

- Your project meets the Logistical, Technical, and Style requirements listed below.
- You turn in the completed Project Proposal on-time (by 10/24).
- You turn in your completed Final Project on-time (by 12/06).

### 2.1 Logistical requirements

- This is a group assignment; but you may not work in groups of more than 2 students.
- There are two deadlines for the final project: 10/24 @ 11:59pm: Project Proposal is due and 12/06 @ 11:59pm: Final Project is due
- NO late submissions will be accepted for either deadline.

## 2.2 Technology requirements

In your final project, you must include the following technologies:

### Frontend: JavaScript

- Write object-oriented JavaScript
- Include 2+ ES6 classes
- Listen for 2+ events; Can be the same event type, like 2 *click* events
- Use *fetch()* to talk to your backend

### Backend

- Use Node and ExpressJS or alternatives (please talk to Dr. Zhang for approval if you plan to use other frameworks)
- Save persistent data in some way using one of the NoSQL databases discussed in class, e.g., MongoDB, Neo4j, Hbase, Amazon S3.
- We encourage you to use a MongoDB backend, but you may use other systems if that is a better fit for your project.
- Include at least 1 GET route
- Include at least 1 POST route, and it should include a message body
- Optional but recommended: include at least one route parameter

### Misc

- Use *async / await* properly somewhere in your project. This can be in the frontend, backend, or both.
- You need to incorporate all these technologies into one cohesive web app. We will not be grading on things like how interesting your project idea is, how original your idea is, how “realistic” it is, etc. However, your project also cannot be a set of totally nonsensical code that happens to hit our checklist of technologies. Your project should be a web app, even if it’s a very small web app, or a silly web app, or a dumb web app.
- Your project should be relatively bug-free. It’s hard for us to judge whether you are using the technology correctly if your code is very buggy. We will not deduct points for trivial bugs or polish bugs, but we will deduct points for things like if the one query you included in your project is not actually working. This is another reason why you should aim small in your project scope.

## 2.3 Style requirements

### JavaScript

- Must write object-oriented JavaScript.
- Must use ES6 classes.
- Avoid global variables. OK to use globals for constants, instantiating classes, or other reasonable scenarios. But you should not put anything in a global variable that could be better encapsulated in a class.

### Backend

- Your backend must be written using the Node and Express libraries or other frameworks approved by Dr. Zhang.
- Don't save data to the filesystem: persistent data should be stored in one of the NoSQL databases.
- HTTP methods should be used in ways that are compatible with the method definition. For example: Use GET for retrieving data. Do not write data in a GET handler. Use POST for saving data. Do not use POST to display a page. Don't use query parameters with POST.

## 2.4 Bonus tasks

The following features are extra credit:

- Support user login using OAuth 2 (<https://oauth.net/2/>). [10%]
- Mobile version. [10%]
- Host the database backend in Amazon AWS. [10%]

## 2.5 Things we are NOT grading

We will NOT grade you on:

- The aesthetics of your web app
- Whether your web app is an interesting app idea
- Whether your web app is a creative app idea
- The technical difficulty of creating your web app
- We strongly recommend you build something small and simple, roughly on par with the scope of the Homework 4.

### 3. Project ideas

You need to decide what web app you will build for the final project. Here are some suggested guidelines for you to follow when deciding what you want to build:

#### **Small and simple**

- You only have a little over 6 weeks for this project, and programmers tend to grossly underestimate how difficult a task is and how long something will take to implement. Try to choose something very focused and simple for the subject of your final project.
- Choose something that feels similar in size and difficulty to Homework 4, or the Freeway app described in Section 4.

#### **No more than 3-5 different pages/screen in the entire app.**

- Homework 4: ecards was 2 screens (creator-view, card-view)
- We recommend you have only 1 to 3 screens in your web app, 5 at the absolute max.
- If your project involves more than 5 screens, it is probably too big in scope for the final project.

#### **If in doubt, build the Freeway app.**

- We've included below a project idea that fulfills the final project requirements: Freeway.
- We will not grade the Freeway app any differently than any other final project.
- If you are having trouble coming up with an appropriate final project idea, build the Freeway app.

### 4. Project ideas: Freeway

The form is titled "Computing the average travel time and total volume for a station". It contains the following elements:

- Input:** Three text input fields labeled "Start date/time", "End date/time", and "Station".
- Submit:** A blue button labeled "Submit".
- Output:** Two text input fields labeled "Travel time" and "Volume".

Figure 1: The compute screen

If you do not want to come up with your own project idea, implement the Freeway app described below.

Figure 2: The update screen

#### 4.1 Freeway app behavior

The Freeway app is a web site that provides a way to query the freeway data available here (<https://gitlab.encs.vancouver.wsu.edu/xuechen.zhang/cs453-project-data.git>). The data you will be using is the files: freeway\_loopdata.csv, freeway\_detectors.csv, freeway\_stations.csv, highways.csv. These files are available in the file ProjectData\_2018.zip. The description of the data can be found in the freeway data set is in the file Freeway\_Data\_Documentation.pdf. Note that the documentation mentions other data files; these other files are not used for this project. The Freeway Data is provided for a section of I-205 NB for a 2-month test period.

#### 4.2 Compute Screen:

When you go to the home page of your Freeway app (shown in Figure 1), you are presented with the compute screen which includes three input text areas, two output text areas, and one “submit” button. When you click the “submit”, your application should compute the travel time and volume using the data from input text areas. Please read Appendix for the method of computing travel times.

- Travel time: Find travel time for station  $x$  (e.g., Foster SB) for a given period of time between start date/time and end date/time. Report travel time in seconds.
- Volume: Find the total volume for the station  $x$  (e.g., Foster SB) for a given period of time between start date/time and end date/time.

#### 4.3 Update Screen:

You should have a second screen in your Freeway app as shown in Figure 2. When the “update” button is clicked, the app will update the names of location text (station name) in your database.

### 5. Proposal and presentation requirements

You will be asked to write a 1-2 paragraph description of your project idea.

- This doesn’t have to be too long or descriptive, and it doesn’t have to be a “pitch”; we just want to get a sense of what you want to make.

- If you are planning to make the Freeway app, you can describe which queries you want to implement and what data you want to display on your browser. Please elaborate if there's anything you plan on changing, though.
- We may reach out to you if we think your project is too difficult or otherwise inappropriate for the final project.

You will be asked to present your proposal in class on 10/29. The minimum requirement of your proposal presentation is to cover the aforementioned items.

## 6. Final presentation requirements

You will be asked to present your final web application on 12/03 or 12/05.

- Demonstrate the execution of your web application.
- If you did the bonus tasks, please consider to visualize the results.
- Critique your data model and project execution. What would you change if you could do the design again?
- Lessons learned: (1) What did you learn about the system and programming framework you used from the project? (2) What would you tell someone who was considering using the system and programming framework you used for the project?

## 7. What You Need to Submit:

For proposals:

- A 1-2 paragraph description
- Names of team members
- Submit it on the Blackboard before 11:59pm on 10/24.

For final project deliverables:

- All source code including Javascript, Html, CSS, and testing programs.
- A subset of your datasets for testing the web application.
- Instructions of running the application.
- Submit the above files as a single zip file (Name: project-<last name>-<WSU\_ID>.zip) before 11:59pm on 12/06 on the Blackboard.

## 8. Appendix: Travel Time Calculation

### 8.1 General Calculation

In general, travel time is calculated by dividing travel length by travel speed. That calculation will give you travel time in hours, multiply by 60 to get travel time in minutes or 3600 to get travel time in seconds.

Travel time in minutes = ((length in miles) / (speed in miles per hour)) \* (60 minutes/hour)

Travel time in seconds = ((length in miles) / (speed in miles per hour)) \* (3600 seconds/hour)

### 8.2 Travel time for a Single Reading

For a single reading (i.e., one row in the freeway\_loopdata file), travel time is length divided by speed, where length is the length of the station associated with the detector in that reading. (A reading contains detectorid, speed, volume and occupancy. You may ignore the dqflags column.)

Travel time in seconds = ((length)/(speed))\*3600

### 8.3 Travel time for a Single Station for a single time period

For a single time period for a single station, travel time is the length of the station divided by the average of the speeds for detectors at that station. You can find associations between detectors and stations in the detectors.csv file.

Travel time in seconds = ((length)/(avg(speed)))\*3600

### 8.4 Travel time for a Multiple Stations for a single time period

To get travel times, for multiple stations (i.e., the I-205 SB Freeway), you calculate the station travel times and then sum the travel times. Note that the formula shows this travel time in minutes and that the multiplier is 60 not 3600.

Travel time in minutes = (sum(((length)/avg(speed))))\*60