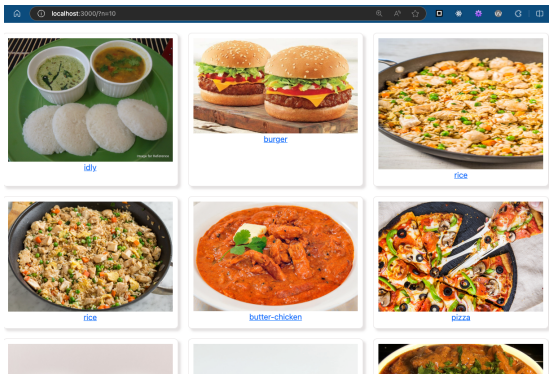
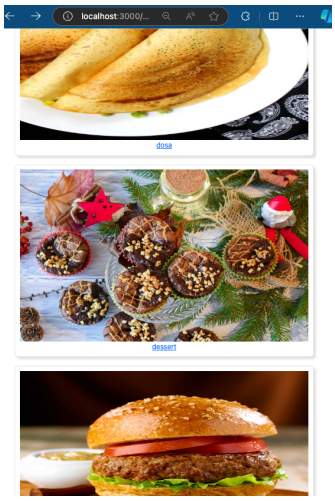
COSC3060 – 2024B – Exercises

Before you start the exercise, please clone the github project
https://github.com/tuanrmit/w7_exercise
which contains the necessary files for the exercise!

Exercise 1.
Let's make a simple web application for retrieving and displaying a random number of food images by accessing the API https://foodish-api.com. The number of images to retrieve is specified in the query part of the URL.

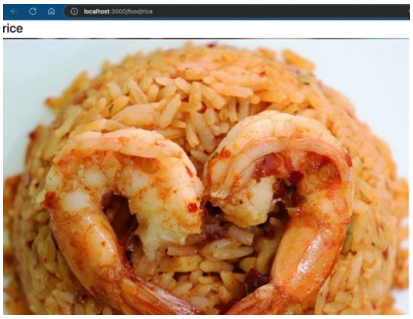


Wireframes for desktop displays (screen width >= 992px)

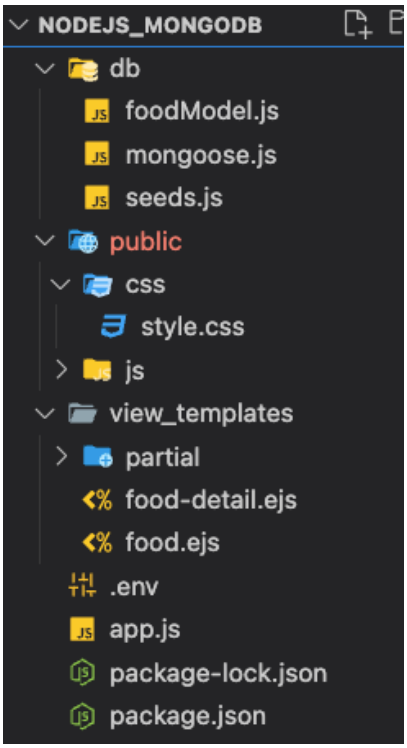


Wireframes for small displays (screen width < 992px)

And clicking the link under the food image will open another tab with random image of the same type of food.



Packages/modules
Install all required packages. Do not use any dependencies other than those predefined in the "package.json".



Database
In the foodModel.js, implement the foodSchema. All data fields are required. Refer to the file seed.js and view its data to know the foodSchema structure
The mongoose.js file helps to connect the app with MongoDB Atlas. Modify the DB environment variable found in the .env file with your MongoDB Atlas connection string details. Make the app connect your MongoDB Atlas, to database of your choice.
Execute the seed.js to insert data into your database.

Server
Modify the .env and configure the app.js so that the port specified in the .env file is used for running the app. It that port is not available, using port 8888 intead.

Modify the code in "app.js" to launch the application. You may need to make some changes to the folder structure to resolve some bugs.

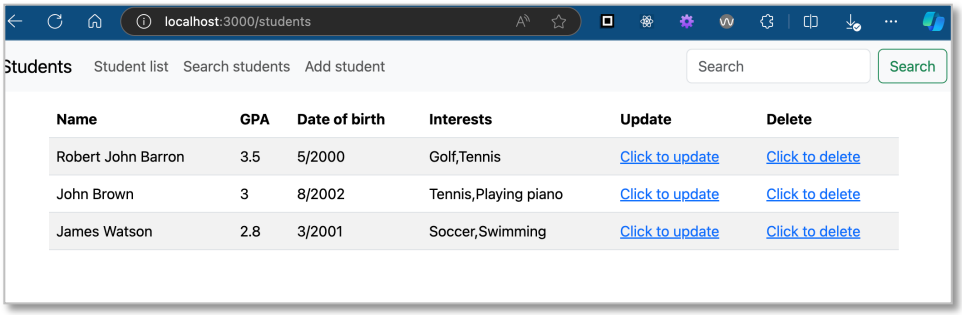
View engine/EJS
Configure the view engine so that .ejs files in the view_templates can be used to display content.

Configuration for convenient use
In the **package.json**, add **two scripts: "start" and "seed".** The "start" script will launch the application, and the "seed" script will insert data into the database, eliminating the need to specify configurations with each execution. These commands will be performed like this:

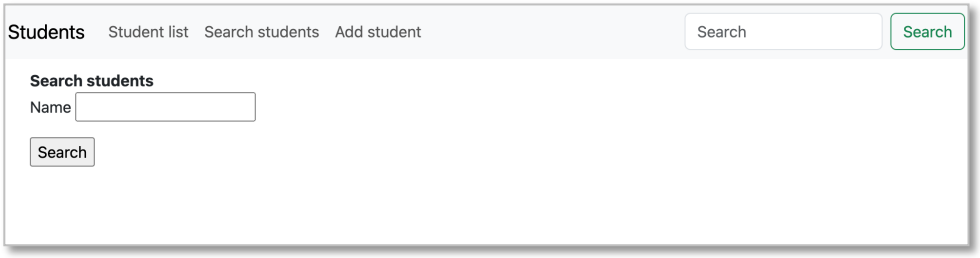
Exercise 2.

Let make a simple application for managing students' information. This application helps to:

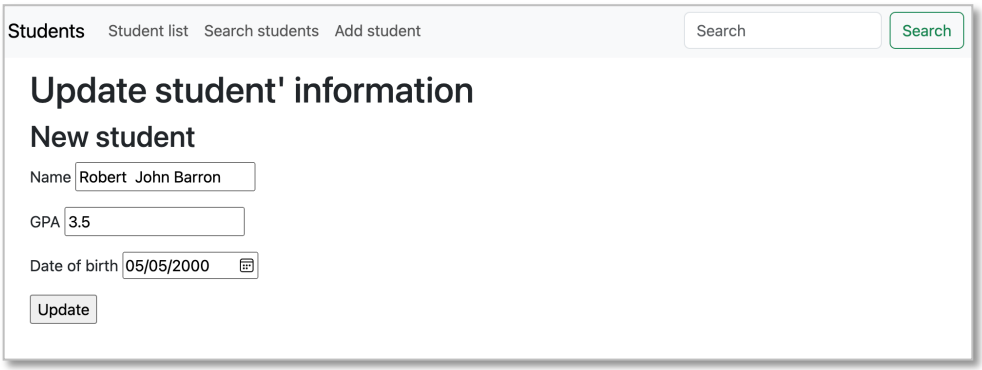
- list all students existing in a database,



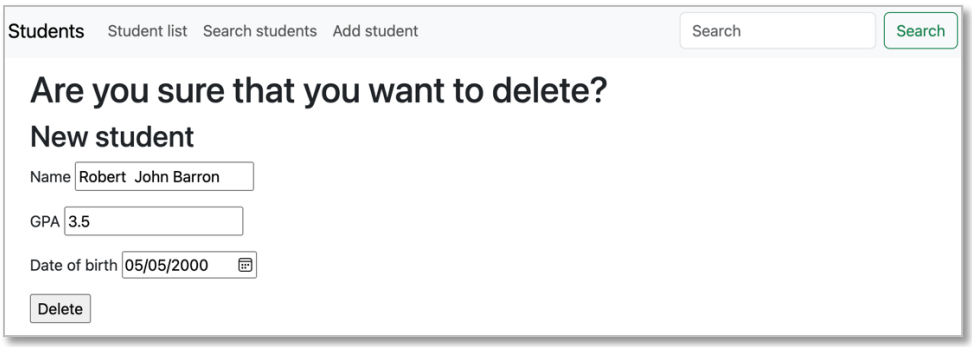
- search for student information by student name,



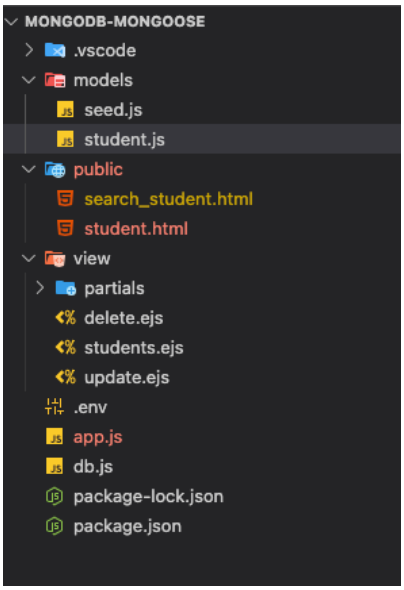
- update students' information,



- and delete student records.



Packages/modules
Install all required packages. Do not use any dependencies other than those predefined in the "package.json".



Database
In the student.js, implement the studentSchema. All data fields are required. Refer to the file seed.js and view its data to know the foodSchema structure
The mongoose.js file helps to connect the app with MongoDB Atlas. Modify the DB environment variable found in the .env file with your MongoDB Atlas connection string details. Make the app connect your MongoDB Atlas, to database of your choice.
Execute the seed.js to insert data into your database.

Server
Modify the .env and configure the app.js so that the port specified in the .env file is used for running the app. It that port is not available, using port 8888 intead.

Modify the code in "app.js" to launch the application. You may need to make some changes to the folder structure to resolve some bugs.

Configure app.js so that static files in public folder is accessable.

View engine/EJS
Configure the view engine so that .ejs files in the view can be used to display content.

https://github.com/tuanrmit/w7_exercise