304CEM

Web API Development

Assignment Brief

You need to develop a full stack, full restful web application. The exact topic you choose is entirely up to you, you can decide to create a news website, a Recipe website, a Cinema website, a portfolio website or any other ideas you are interested in.

You will need to do four tasks:

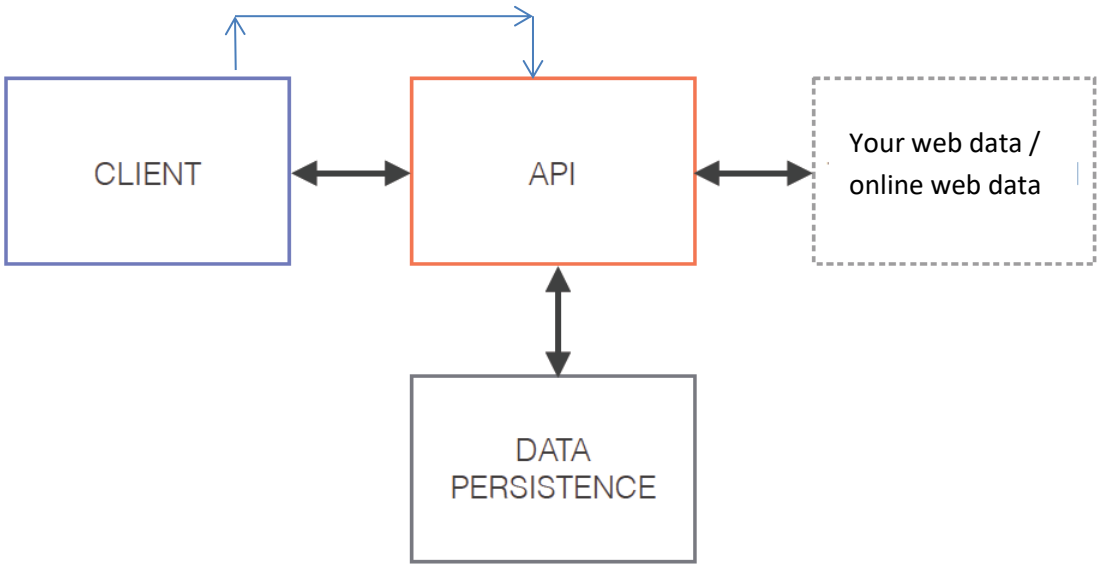
1- Create a persistent data storage (SQL, noSQL, JSON files, text files or any other technology you think suitable for your website) and write one page report about its design 2- A backend system in a form of a full Restful public web API using NodeJS, in case you are not comfortable with NodeJS you may only use Python, no other alternatives are permitted. For this part of the assignment the source code should be hosted on a GIT repository 3- A web application using combination of modern web UI technologies such as Bootstrap and JQuery. The front-end should communicate with your backend system, the source code should hosted on a GIT repository. 4- A link to your live application or a video recording of your application.

Task 1: Persistent Storage (20%)

You should create a persistent data storage, you can choose the technology you think will be most suitable for your application, it can be a SQL database (e.g. mySQL, SQL Server), a noSQL database (e.g. MongoDB, CouchDB) a files system, text files or any other technology. You should explain your database design and justify the design and the chosen technology in a one page report and submit it through moodle.

Architecture

The diagram below may help you understand the requirements. Notice that the client interacts with your API **OR** directly interact with the third-party API(s)s and with your data persistence solution.



Task 2: Backend System (Public Web API) (35%)

You should create a backend system for your application, this backend system should be in a form of a full Restful public web API, you will learn how to create such system during this module, your backend system should be written in NodeJS only, you can use any framework or library available, so for example you may use NodeJS + RESTIFY. Please note that you will get a zero grade if you choose other languages. Your backend system should serve at least the following functionality:

1- User Registrations
2- Login System and Authentication.
3- At least two CREATE methods using POST requests to insert data in the database (apart from user registration or login activities)
4- At least two UPDATE methods using PUT requests to update data in the database (apart from user registration or login activities)
5- At least two DELETE methods using DELETE requests to delete data from the database (apart from user registration or login activities)
6- At least two RETRIEVE methods using GET requests (apart from user registration or login activities)

For example, you may suggest developing a Recipe website, then your minimum requirements could be:

1- A public API for user's registrations
2- A public API for user's login
3- Public APIs to add a recipe, and add an existing recipe to a favourite list(2 CREATE methods)
4- Public APIs to update a recipe, a comment, and a specific user's profile information (2 UPDATE methods)
5- Public APIs to delete a recipe, and a specific recipe from a specific user's favourite list (2 DELETE methods).
6- Public APIs to retrieve a recipe, and a specific user's favourite list. (2 RETREIVE methods).

You should submit a link to your GIT repository.

Task 3: The front-end System (35%)

You will need to prototype, design, and implement a single page application for your project using a mixture of HTML5, Vanilla Javascript, Bootstrap, JQuery. For more extra credits and challenge you may use Single Page Application frameworks such as AngularJS or Vue.JS. You should not use any other front-end system frameworks other than what is proposed above, the reason for that, is to make marking and assessment consistent. The web application should connect to the back-end server to read and write data. This application should at least allow the user to register, login, and perform the other required 12 operations (2 creates, 2 updates, 2 deletes, 2 retrieves).

You will be assessed on the design of your front-end and its, completeness, effectiveness and

Task 4: Video Recording (10%)

You can evidence your work with either submitting a link to your live application **or** submitting a video recording of it.

1. Live demonstration link **OR**
2. A link to your reflective video (hosted on YouTube)

Video

Once you have completed the API and the client, you need to explain how your API and client work. Rather than writing a report, you are required to record a short screencast of 8 min or less. This should cover all the points in the grading criteria and demonstrate your skills and knowledge of the subject. After uploading, you must change the video permissions from private to **unlisted**. This will allow the lecturers to access the video via the link but prevent it being publicly searchable.

Make sure you separately demonstrate the API (using cURL or Postman) and the client and ensure you justify your choices of language constructs and architecture.

1. Demonstrate your API showing the requests and responses (headers/body)
2. Demonstrate the back-end persistence showing how the data is stored
3. Run the unit and acceptance tests on the API explaining code coverage
4. Demonstrate the features of the web client even **it is not functioning**
5. Run the unit and acceptance tests on the web client explaining code coverage
6. Video narration must be done by **yourself.** For any suspected voice noticed, marks will be deducted.

Final Submission

This assignment requires you to demonstrate the range of skills and knowledge required by industry and this is reflected in the coursework submission. You are required to submit the following. 1. Links to your Git remote repositories (hosted on University Github), you will have a repository for your backend API and a repository for your front-end. 2. A link to your live application where all components working together (persistent storage + API + Front-end) or a link to your reflective video (hosted on YouTube) 3

Source Code

Managing source code is a vital skill if you are to become a successful developer. For this assignment, you are required to track your API code and your client code in separate Git repositories and you will be marked on how efficiently you organise this.

You will be required to submit links to both your Git remotes hosted on Github. Both repositories should include full documentation available through the home page.

To ensure the code can be seen by your lecturers, make sure you give them **reporter** permission. Their usernames are:
alexng88

Grading

- 35% Front-end
- 35% Back-end
- 20% Persistent Storage
- 10% Video Demo

Marks will be deducted on late submission.

Deadline

06 April 2020

1 week	Your marks x 90%
2 weeks	Your marks x 80%
More than 2 weeks	Your marks x 0%