UNIVERSITÉ Concordia UNIVERSITY CONTINUING EDUCATION

MODX: Advanced Web Programming

Instructor: Marcos Sanches

Instructor Email: marcos.sanches@concordia.ca

Course prerequisite(s): TI-402 Web Programming

Course Description: This course builds on the knowledge of TI-402 (Web Programming) to go deeper into the areas of HTML, CSS, XML, JavaScript and other Web technologies focusing on the server-side coding and database manipulation required for enterprise-level web applications. In this course, students learn how to use a modern front-end framework, Vue.js, as well as Node.js as a back-end framework. They also learn how to make a client and server communicate using a RESTFUL JSON API.

Learning Outcomes:

By the end of this course, students will be able to:

1. Develop software using a client-server architecture.
2. Implement a front-end software solution using Vue.js.
3. Implement a back-end software solution using Node.js.
4. Apply component web-development architecture.
5. Understand the web API architecture and it's applications.
6. Create front-end routes.
7. Create back-end routes.
8. Understand and use the HTTP protocol within web API's.

Context

1. Project problem statement.
2. Project Description.
3. Work breakdown Approach.
4. Deliverable documentation format (Milestones).
5. References.

Description

The project for this course is intended to run throughout the course and to be completed by the last day of the course. During this time, several checkpoints should be made to ensure that students are in fact starting the work and are not in trouble or stuck on any technical challenges.

- The project is worked on within a group of 4 students.
- The project is meant to practice development using the topics learned in the course.
- The project also encourages group work and group coordination using tools such as GIT.
- Development is meant to be done in phases or parts, and this should start very close to the beginning of the course. There will be a presentation of the group's project before the end of the course.
- Participation within the group will also be graded and evaluated to ensure a fair distribution of work among team members.

Assignment/Project submission:

Assignments and project milestones must be submitted via the course management system (Moodle) on the due date. Labs must be started and done during lab times. In the case that a deliverable must be continued outside of class, a student must submit their partial work and later, the final product. Please note that late submissions may be subject to a grading penalty of 10% per day. If you foresee an issue with a submission date, please communicate with the instructor well before the due date.

1. Project problem statement.

Develop a to do list with 3 status: To do, doing, done.

To access the to do list the user must pass through a login page.

All pages must be secure using a token to store the session.

2. Project Description.

Frontend Layer
A front-end designed with Vue.js, using the traditional Model-View-Controller (MVC) or Model-View-View-Model (MVVM) architecture.

API / Service Layer
A service should be created to support all front-end requests, but it must not stop there. It should be designed in a flexible way to support future requirements.

Use of time-based tokens for the authentication process. Using the token for all subsequent requests.

Important: A public web server should be used to host the service / database of the application.

Business Layer / Entity Layer

The business layer serves the purpose of storing the entity definitions and providing user-accessible methods as an abstracted interface to the application's data. IT should be easy to use, it should make use of business objects, and most importantly it should provide the validation necessary to ensure clean data.

Database Layer
The database layer exists to persist data.

The database to be used is MySQL.

3. Work breakdown approach.

The project is broken down into four Milestones. For each week of the course duration.

Milestone: Each milestone is considered to be an assignment in itself which should adhere to the late policies of the course

Milestone 1 : Documentation and Design. (20%)

1. Approval of the project structure and design (the theme, or purpose also).
2. Project design (UML charts, structure charts, etc…)
3. Wireframe examples of the front-end (with the understanding they may change).
4. Initial design of database tables required.
5. Work not completed at this milestone would be penalized.

Milestone 2: Prototype (20%)

1. The front-end should be "mostly" functional at this point.
2. The API should also be functional with the required HTTP methods in place to allow the front-end to work.
3. The back-end should include the business layer and entities. But the data tier is optional. Data can be "mocked" at this point using non-persistent data such as lists, dictionaries or hashmaps.

Milestone 3: Completed Project and Presentation (40%)

1. The project is complete and functional. Nothing is left to do.
2. Security must be implemented into the API/Frontend at this point. No requests should be allowed to happen to the web service without proper authorization in place.

Presentation 5 - 7 min. (20%)

Explaining the project (front-end and backend)

Brief explanation of challenges encountered and how they were resolved.

4. Deliverable Format.

Documentation and Design

For each deliverable, documentation is important.

1. User relevant images, charts for more explanation.

2. Submit it on portal late submission will not be evaluated.

3. Documentation must include:

- Cover page with student names.
- Brief introduction of the project explaining the main objective and the technologies that will be used during the development.
- Brief description of the role of each student during the development of the project
- Project plan divided into 3 sprints (one for each milestone) with the description of the backlog, the resources needed and the effort (in time).
- Presentation and explanation of all graphs, diagrams, and models.

4. Propre documentation: Font size: 11px.

- Font: Arial.
- Heading: Bold (if necessary).
- Number of pages: Not more than 5 pages. **(Depends).
- Include references: url of website. Evaluation, done on all the above criteria. Follow the evaluation process.

Prototype

Project file compressed in ZIP format, including database SQL file (data and structure).

Completed Project and Presentation

Compressed folder in ZIP format, including:

- Final prototype files with new changes
- Final database SQL file (data and structure).
- Final documentation with all changes done during the project.
- Presentation file.

Not following the deliverable format, has a penalty on the milestone grading.

5. References.

It includes the source from which you have gathered the ideas and knowledge to implement during each deliverable. Include URLs.