JOHN ABBOTT CEGEP/COLLEGE COURSE OUTLINE

A. GENERAL INFORMATION

1. Program name:	Computer Science Technology 420.B0
2. **Course title:**	**Web Programming II**
3. Course and section number:	420-4W6-AB Section 2
4. Ponderation (weekly class –lab/fieldwork – homework hours):	2 - 4 - 3
5. Credits:	3.00
6. Competency statements(s) and code(s):	00SU - Develop transactional Web applications. Covered: Partially
7. Prerequisites:	420-3W5-AB, 420-3D6-AB
8. Semester:	Winter 2025
9. Teacher name:	PreetiPearl D'Melo
10. Office number, phone extension:	P-311
11. Teacher's availability:	Posted on office door.

B. INTRODUCTION

Course summary

By the end of this course, the student will be able to fully develop a transactional web application. After mapping out the requirements of the business logic, the web apps will be organized using Model-View-Controller architecture. The application’s state will be persisted using a database. They will be able to handle HTTP requests using a REST API. The code will be robust by having the student evaluate the web application’s quality through software testing. The course will be taught using PHP and/or NodeJS.

Role of the course

This course is a prerequisite for 420-5W6-AB and 420-5N4-AB.

C. COURSE OBJECTIVES

Upon successful completion of this course, students will be able to:

General Learning Objective	Specific Learning Objective
1. Handle errors and exceptions.	1. Enable different levels of error reporting. 2. Differentiate the main types of errors. 3. Define exception handling. 4. Differentiate errors and exceptions. 5. Handle exceptions using try/catch blocks. 6. Assess potential application-specific problems for which to write custom exceptions.
2. Organize a web application using Model-View-Controller architecture.	1. Explain Model-View-Controller architecture. 2. Use a template engine to encapsulate all view-related code. 3. Transact data to and from a database using models. 4. Interface between the views and models using controllers.
3. Route HTTP requests using a REST API.	1. Define REST and for what it is used. 2. Rewrite URLs to conform to REST protocols. 3. Route to the proper controller based on the HTTP method and resource requested.
4. Evaluate web application quality through testing.	1. Explain why software testing is important. 2. List the different types of software tests. 3. Compare and contrast the main types of software tests. 4. Write unit tests for a web application. 5. Determine whether the application has been thoroughly tested by looking at the code coverage.

General Learning Objective	Specific Learning Objective
5. Map out the requirements of a web application.	1. Outline the software development life cycles. 2. Collect information on the number & type of users, degree of interactivity of the application, and the information distribution mode. 3. Deduce and justify the application's use cases. 4. Write an analysis report based on the gathered requirements. 5. Produce a web application proposal prototype. 6. Write documentation for a web application.
6. Develop a transactional web application.	1. Outline the web request-response cycle. 2. Set up a development environment configured for coding a web application. 3. Load text, images, and data asynchronously to update a web page. 4. Manage application state with sessions & cookies. 5. Authenticate users to access the web application. 6. Determine when to use GET, POST, PUT, and DELETE requests.

Government Competencies

Competency 00SU. Develop transactional Web applications

Achievement Context

- For transactional Web applications: reservations, registrations, collaboration, inventory management, e-commerce, etc.
- For new applications and applications to be modified
- Based on design documents
- Using images
- Using issue tracking and version control procedures

Elements of competency	Performance criteria
00SU.1 Analyze the application development project.	• Accurate analysis of design documents • Proper identification of the tasks to be carried out

Elements of competency	Performance criteria
00SU.2 Prepare the computer development environment.	• Proper installation of the Web development platform and the development database management system • Proper installation of software and libraries • Appropriate configuration of the version control system • Proper importing of the source code
00SU.3 Prepare the database.	• Suitable creation or adaptation of the database • Proper insertion of initial or test data • Compliance with the data model
00SU.5 Program the server-side application logic.	• Proper programming or integration of authentication and authorization mechanisms • Proper programming of interactions between the Web interface and the user • Appropriate choice of clauses, operators, commands or parameters in database queries • Correct handling of database data • Appropriate use of data exchange services • Proper application of internationalization techniques • Precise application of secure programming techniques
00SU.7 Control the quality of the application.	• Precise application of test plans • Thorough reviews of code and security • Relevance of the corrective actions • Compliance with issue tracking and version control procedures • Compliance with design documents
00SU.8 Participate in the deployment of the application on the Web host.	• Accurate identification of the domain name • Appropriate configuration of the application on the Web host • Proper application of the procedure for migrating the service onto the Web host • Precise application of security measures • Compliance with search engine indexing requirements

Elements of competency	Performance criteria
00SU.9 Produce the documentation.	• Proper identification of the information to be written up • Clear record of the work carried out

D. EVALUATION PLAN

Evaluation type:	%	Tentative date:	Competencies/Objectives	Part of the final evaluation?
Assignments (4-6)	35% (each <= 9%)		00SU.1 00SU.2 00SU.3 00SU.5 00SU.7	√
Test 1	20%	Week 6	00SU.1 00SU.3	√
Test 2	20%	Week 12	00SU.1 00SU.3 00SU.5	√
Project	25%	Week 15	00SU.1 00SU.2 00SU.3 00SU.5 00SU.7 00SU.8 00SU.9	√
Total value:	*100%*			
Value of final eval (min 40%)	*100%*			

E. COURSE CONTENT

Tentative Schedule

Week	Topic
1	Node JS Basics and Development Environment Setup
2	NodeJS Basics and MVC
3	Models and Error Handling
4	Models and Unit Testing
5	HTTP Request and Response
6	Controllers and Routing, **Test 1**

Week	Topic
7	REST API and Endpoint Testing
8	Logging
9	Views
10	Views
11	Navigation
12	Cookies and Project, **Test 2**
13	Authentication and Project
14	End to End Testing and Project
15	Project

F. REQUIRED TEXTBOOKS/MATERIALS

TBD

G. BIBLIOGRAPHY

Not applicable.

H. INSTRUCTIONAL METHODS

This course consists of 90 hours of scheduled lectures and lab work. In addition, each student will be required to do 45 hours of personal study that includes research, personally booked computer time, and work at home. Léa, the course management system within Omnivox, will be used in this course.

I. PROGRAM, DEPARTMENTAL/DISCIPLINE, AND COURSE/SECTION POLICIES

Topic	Resource
Approved department attendance policy	Active learning of competencies requires hands on learning with interactive classroom work, which cannot be achieved with absences of more than 20%. Attendance is mandatory for the following program activities: • **Stage courses:** Absences that amount to more than 20% of the hours associated with the stage activity may mean that a student cannot achieve more than 50% on the stage course, resulting in the failure of the stage course. • **Stage evaluation meetings:** Absences that amount to more than 20% of the hours dedicated to these meetings will result in a grade of zero on this portion of the stage course, unless the absences have been deemed excused absences as per the IPESA. • **In class group work or project integration activities:** Absences that amount to more than 20% of the hours associated with any of these activities will result in a grade of zero on the assessment related to the missed activity, unless the absences have been deemed excused absences as per the IPESA.
Policy to ensure that issues relating to late submission, or resubmission, or work to be dealt with in an equitable manner	All assignments and projects are expected to be submitted by the required due date. A late penalty of between 5 to 10% per day might apply to assignments submitted late up to a maximum number of allowed late days. The exact percentage and the maximum will be indicated on the instructions specific to that assignment. Any work submitted after the maximum number of allowed late days has been reached will not be graded.

Topic	Resource
Policy dealing with the expectations of classroom behaviour, including use of cell phones, laptops and other technology	Not applicable.

J. COLLEGE POLICIES

Topic	Resource
Student rights and responsibilities (articles 3.2 and 3.3) Changes to evaluation plan in the course outline (article 5.3) Religious holidays (articles 3.2.13 and 4.1) Cheating and plagiarism (articles 9.1 and 9.2)	Policy 7:IPESA - Institutional Policy on the Evaluation of Student Achievement (version: June 12, 2019)
Cheating and plagiarism academic procedure and other resources	Academic Integrity: Cheating and Plagiarism Procedure (version: October 22, 2021) • **You need to log into Omnivox to access the above document** • **For PowerPoint on cheating and plagiarism** refer to the JAC Portal: My JAC Communities / Academic Council / Curriculum Validation Committee (CVC) / Course Outlines -- Reference Documents / Academic Integrity • For link to interactive tutorial on how to cite sources correctly: http://citeit.ccdmd.qc.ca
Code of conduct	Policy 13: Policy on Student Conduct and Discipline Procedures (version: September 21, 2021)