



Course Title:	Full Stack Web Development										
Course Code:	CSFS 187										
Subject:	Computer Science										
Credits:	4										
Contact Hours:	50										
Semester/Term:	<input checked="" type="checkbox"/> Semester <input type="checkbox"/> J-Term <input type="checkbox"/> Summer										
Course Description:	<p>The course explores complex database integration and utilizes various servers to create full-stack web applications, and builds on the front-end web application development skills from CSFE 185 and CSIS 186. The final portion of the course is dedicated to building full-stack web applications with persistent data in project groups. The web applications created in this course serve as the foundation for a professional portfolio.</p> <p>Throughout the course, students will more broadly gain an appreciation and framework for learning new technologies and in understanding broader software development principles. Students also learn teamwork, project management and presentation skills as part of a collaborative final project to develop a complex web application that integrates prior learning.</p>										
Course Requirements:	<p>Required Text Digital access and/or copies will be provided at no cost to students.</p> <p>Assignments</p> <table border="1"> <thead> <tr> <th>Course Requirements</th> <th>Percentages</th> </tr> </thead> <tbody> <tr> <td>Employee Tracker System</td> <td>20 %</td> </tr> <tr> <td>Food Builder Application</td> <td>20 %</td> </tr> <tr> <td>Full-Stack Web Application Project/Presentation</td> <td>60%</td> </tr> <tr> <td>Total</td> <td>100%</td> </tr> </tbody> </table> <p>Employee Tracker System (20%) Build an employee registry command line app using Node, Inquirer, and MySQL that allows users to create employees and assign roles, salaries, departments, and direct reports and utilizes multiple tables, JOINS, and GROUP BY statements.</p> <p>Food Builder Application (20%) Create an application that allows the user to dynamically add to a list and logs a boolean state for each item, accessing rich content stored in MySQL servers, and using the Model-View-Controller paradigm.</p> <p>Full-Stack Web Application Project (60%)</p>	Course Requirements	Percentages	Employee Tracker System	20 %	Food Builder Application	20 %	Full-Stack Web Application Project/Presentation	60%	Total	100%
Course Requirements	Percentages										
Employee Tracker System	20 %										
Food Builder Application	20 %										
Full-Stack Web Application Project/Presentation	60%										
Total	100%										



	Complete the development of a complex full-stack web application with a project team. Prepare and deliver a professional presentation as a project team.		
Learning Outcomes and/or Expected Student Competencies and Assessment Measures:	On completion of the course, students should be able to:		
	Learning Outcomes	Course Requirement that will be used to assess the student's achievement of this outcome	
	Configure to a MySQL instance; Create and drop databases and tables; Explain and execute CRUD methods; Use both a GUI and the MySQL command prompt to interface with a database; Create schema and seed files for development and production; Write join statements to establish relations between data in one or more tables	Employee Management System	
	Build a full-stack web application; Configure a full-stack web application to implement CRUD methods using the Sequelize ORM library; Implement user authentication with Passport.js; Create complex database relationships; Develop migration files to ensure consistency among databases in collaborative environments	Blog	
	Render dynamic HTML using the Handlebars.js template engine; Explain and implement MVC modularization in a full-stack web application and a custom ORM in the Model layer of an MVC framework; Configure Heroku for deployment of an application using MySQL; Explain the interaction between the model, view and controller; Explain the benefits of object-relational mapping	Restaurant App	
Explain the difference between SQL and NoSQL; Configure Heroku for deployment of a Node.js application using MongoDB; Explain and execute CRUD methods with MongoDB, using both the and Mongoose.js; Use both a GUI and the Mongo command prompt to interface with a database; Integrate Mongoose in a full-stack web	Budget Tracker v1		



	<p>application; Create query builders to populate documents using refs; Implement client-side NOSQL storage using IndexedDB</p>																											
	<p>Design and develop user stories for testing apps; Conceptualise full stack apps; Delegate responsibilities for app development based on a project's requirements; Formulate effective presentations that communicate complex full stack apps</p>	<p>Full-Stack Web Application Project</p>																										
<p>Course Outline:</p>	<table border="1"> <thead> <tr> <th data-bbox="464 705 695 747">Session</th> <th data-bbox="695 705 1468 747">Topic</th> </tr> </thead> <tbody> <tr> <td data-bbox="464 747 695 810"><i>Session 1</i></td> <td data-bbox="695 747 1468 810">MySQL, Database Management</td> </tr> <tr> <td data-bbox="464 810 695 873"><i>Session 2</i></td> <td data-bbox="695 810 1468 873">MySQL, Database Management</td> </tr> <tr> <td data-bbox="464 873 695 936"><i>Session 3</i></td> <td data-bbox="695 873 1468 936">Sequelize ORM, Database Models and Associations, Passport.js</td> </tr> <tr> <td data-bbox="464 936 695 999"><i>Session 4</i></td> <td data-bbox="695 936 1468 999">Sequelize ORM, Database Models and Associations, Passport.js</td> </tr> <tr> <td data-bbox="464 999 695 1062"><i>Session 5</i></td> <td data-bbox="695 999 1468 1062">Sequelize ORM, Database Models and Associations, Passport.js</td> </tr> <tr> <td data-bbox="464 1062 695 1125"><i>Session 6</i></td> <td data-bbox="695 1062 1468 1125">Connecting Express with MySQL, Semantic Templating with Handlebars, The Model-View-Controller Paradigm, Intro to ORMs</td> </tr> <tr> <td data-bbox="464 1125 695 1188"><i>Session 7</i></td> <td data-bbox="695 1125 1468 1188">Connecting Express with MySQL, Semantic Templating with Handlebars, The Model-View-Controller Paradigm, Intro to ORMs</td> </tr> <tr> <td data-bbox="464 1188 695 1251"><i>Session 8</i></td> <td data-bbox="695 1188 1468 1251">MongoDB, The Mongoose ORM, IndexedDB</td> </tr> <tr> <td data-bbox="464 1251 695 1314"><i>Session 9</i></td> <td data-bbox="695 1251 1468 1314">MongoDB, The Mongoose ORM, IndexedDB</td> </tr> <tr> <td data-bbox="464 1314 695 1377"><i>Session 10</i></td> <td data-bbox="695 1314 1468 1377">MongoDB, The Mongoose ORM, IndexedDB</td> </tr> <tr> <td data-bbox="464 1377 695 1440"><i>Session 11</i></td> <td data-bbox="695 1377 1468 1440">Collaborative Full-Stack Development</td> </tr> <tr> <td data-bbox="464 1440 695 1503"><i>Session 12</i></td> <td data-bbox="695 1440 1468 1503">Collaborative Full-Stack Development</td> </tr> </tbody> </table>		Session	Topic	<i>Session 1</i>	MySQL, Database Management	<i>Session 2</i>	MySQL, Database Management	<i>Session 3</i>	Sequelize ORM, Database Models and Associations, Passport.js	<i>Session 4</i>	Sequelize ORM, Database Models and Associations, Passport.js	<i>Session 5</i>	Sequelize ORM, Database Models and Associations, Passport.js	<i>Session 6</i>	Connecting Express with MySQL, Semantic Templating with Handlebars, The Model-View-Controller Paradigm, Intro to ORMs	<i>Session 7</i>	Connecting Express with MySQL, Semantic Templating with Handlebars, The Model-View-Controller Paradigm, Intro to ORMs	<i>Session 8</i>	MongoDB, The Mongoose ORM, IndexedDB	<i>Session 9</i>	MongoDB, The Mongoose ORM, IndexedDB	<i>Session 10</i>	MongoDB, The Mongoose ORM, IndexedDB	<i>Session 11</i>	Collaborative Full-Stack Development	<i>Session 12</i>	Collaborative Full-Stack Development
Session	Topic																											
<i>Session 1</i>	MySQL, Database Management																											
<i>Session 2</i>	MySQL, Database Management																											
<i>Session 3</i>	Sequelize ORM, Database Models and Associations, Passport.js																											
<i>Session 4</i>	Sequelize ORM, Database Models and Associations, Passport.js																											
<i>Session 5</i>	Sequelize ORM, Database Models and Associations, Passport.js																											
<i>Session 6</i>	Connecting Express with MySQL, Semantic Templating with Handlebars, The Model-View-Controller Paradigm, Intro to ORMs																											
<i>Session 7</i>	Connecting Express with MySQL, Semantic Templating with Handlebars, The Model-View-Controller Paradigm, Intro to ORMs																											
<i>Session 8</i>	MongoDB, The Mongoose ORM, IndexedDB																											
<i>Session 9</i>	MongoDB, The Mongoose ORM, IndexedDB																											
<i>Session 10</i>	MongoDB, The Mongoose ORM, IndexedDB																											
<i>Session 11</i>	Collaborative Full-Stack Development																											
<i>Session 12</i>	Collaborative Full-Stack Development																											
<p>Other Policies:</p>	<p>Expectations Professional behavior is expected of all students. This includes preparation for classes, on-time attendance at classes, attendance at all group sessions and appropriate participation in the form of attentiveness and contributions to the course. Respect for the academic process is the major guiding principle for professional behavior and extends to all communications, including email.</p>																											



	<p>Attendance/Participation Prompt attendance, full preparation, and active participation in class discussions are expected from every student in every class session. Students who miss more than two course sessions will be in jeopardy of successful completion of the course and its requirements. Students should consult their instructor regarding absences to address missed work and/or make-up assignments.</p> <p>Course Policies For email communications, students must use their Arcadia University email account. Students are responsible for any information provided by e-mail or through Intranet postings.</p> <p>Plagiarism Representation of another's work or ideas as one's own in academic submissions is plagiarism, and is cause for disciplinary action. <i>Cheating</i> is actual or attempted use of resources not authorized by the instructor(s) for academic submissions. Students caught cheating in this course will receive a failing grade. <i>Fabrication</i> is the falsification or creation of data, research or resources to support academic submissions, and cause for disciplinary action.</p> <p>Late or Missed Assignments Will not be accepted for grading. Completion of all projects and homework is required for course completion.</p> <p>Students with Disabilities Persons with documented disabilities requiring accommodations to meet the expectations of this course should disclose this information while enrolling into the program, and before the start of term so that appropriate arrangements can be made.</p> <p>Title IX Statement Arcadia University is committed to assuring a safe and productive educational environment for all students. In order to meet this commitment and to comply with Title IX of the Education Amendments of 1972 and guidance from the Office for Civil Rights, the University requires faculty members to report incidents of sexual violence shared by students to the University's Title IX Coordinator. The only exceptions a faculty member's reporting obligation are when incidents of sexual violence are communicated by a student during a classroom discussion, in a writing assignment for a class, or as part of a University-approved research project.</p> <p>Information regarding the reporting of sexual violence and the resources that are available to victims of sexual violence is set forth at https://www.arcadia.edu/university/policies-guidelines/title-ix.</p>
Prerequisites:	CSFE 185, CSIS 186



ARCADIA UNIVERSITY
THE COLLEGE OF GLOBAL STUDIES

Intellectual property copyright Arcadia University