College of San Mateo
Official Course Outline

1. **COURSE ID:** CIS 129 **TITLE:** Frameworks/Server-Side JavaScript
   Units: 3.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; 96.0-108.0 Homework hours; 144.0-162.0 Total Student Learning hours
   Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
   Recommended Preparation:
   Completion of CIS 111.

2. **COURSE DESIGNATION:**
   Degree Credit
   Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**
   Catalog Description:
   Introduction to JavaScript frameworks. Provides an overview of MVC (Model-View-Controller), MVP (Model-View-Presenter) and MVVM (Model-View-ViewModel) design patterns. Server-side JavaScript programming with package managers will be introduced. Also covered are REST (REpresentational State Transfer) Web Services and APIs used to obtain and process data in XML or JSON format. Overview of JavaScript templating engines for Web applications. Various open-source libraries for DOM manipulation, GUI, visualization and testing will be introduced. Intended for students with previous programming experience.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**
   Upon successful completion of this course, a student will meet the following outcomes:
   1. Develop interactive Web applications using JavaScript frameworks.
   2. Develop interactive Web applications that integrate client- and server-side JavaScript programming.
   3. Employ REST APIs to fetch XML or JSON data asynchronously from the server.
   4. Explain JavaScript MVC, MVP and MVVM design patterns and illustrate how they are used to create various applications.
   5. Create an advanced project using various libraries and frameworks, with attention to security and performance.

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**
   Upon successful completion of this course, a student will be able to:
   1. Develop interactive Web applications using JavaScript frameworks.
   2. Develop interactive Web applications that integrate client- and server-side JavaScript programming.
   3. Employ REST APIs to fetch XML or JSON data asynchronously from the server.
   4. Explain JavaScript MVC, MVP and MVVM design patterns and illustrate how they are used to create various applications.
   5. Create an advanced project using various libraries and frameworks, with attention to security and performance.

6. **COURSE CONTENT:**
   Lecture Content:
   1. Introduction
      A. Model-View-Controller Architecture
      B. Model-View-Presenter Architecture
      C. Model-View-ViewModel Architecture
      D. JavaScript on the Server
      E. Debuggers
   2. Framework Basics
      A. Introduction to MV* Patterns
      B. How to Choose a Framework
      C. Common Frameworks
         a. React
         b. Backbone
         c. Angular

d. Ember
e. Vue
f. Meteor

3. Server-Side JavaScript
   A. Introduction to Node.js
   B. Command-line Interface
   C. Package Manager npm
   D. Node Modules
4. Web Services
   A. REST Architecture
   B. Web APIs
   C. JSON Data Format
   D. XML Data Format
5. Templating Engines
   A. Introduction to Templates
   B. Common Templating Engines
      a. Mustache
      b. Handlebars
      c. EJS
      d. Underscore
      e. jTemplates
   C. Real-World Examples
6. JavaScript Libraries
   A. Common DOM-based Libraries
      a. MooTools
      b. jQuery
      c. Dojo
      d. Prototype
   B. Visualization Libraries
      a. D3.js
      b. Highcharts
      c. p5.js
      d. Three.js
   C. GUI-related Libraries
      a. Bootstrap
      b. Dojo Widgets
      c. jQuery UI
      d. Polymer
      e. EXT JS
   D. Unit Testing Libraries
      a. Jasmine
      b. Mocha
      c. QUnit
      d. Unit.js
7. Integrating the Client and Server
   A. Client-Side Code
   B. Server-Side Code
   C. Browser Issues
8. Security and Performance
   A. JavaScript and Browser Security
   B. Communicating with Remote Services
   C. Protecting Confidential Data
   D. Restricting Access to Web Data

7. **REPRESENTATIVE METHODS OF INSTRUCTION:**

Typical methods of instruction may include:

A. Lecture
B. Activity
C. Discussion
D. Other (Specify): Teacher will model problem-solving techniques. Class will solve a problem together, each

person contributing a potential "next step". Teacher will create and manage an Internet conference for discussion of course topics. Students will work in small groups to solve programming problems.

8. **REPRESENTATIVE ASSIGNMENTS**

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

Documentation for programming assignments

Reading Assignments:

Reading assignments accompanied by self-test questions and running and testing code samples. Posted lecture notes and relevant handouts. Documentation for framework, library, templating engines, and Web Service APIs.

Other Outside Assignments:

A. Web applications using various JavaScript frameworks.
B. Web applications using Node.js on the server-side.
C. Web applications using various JavaScript libraries.
D. Web applications using various templating engines.
E. Unit testing on Web applications.

9. **REPRESENTATIVE METHODS OF EVALUATION**

Representative methods of evaluation may include:

A. Class Participation
B. Class Work
C. Exams/Tests
D. Homework
E. Lab Activities
F. Projects
G. Quizzes
H. Written examination
I. Bi-weekly quizzes (short answer--from textbook material) to provide feedback to students and teacher; Assessment of student contributions during class discussion and project time; Individual programming assignment; Midterm and Final exams, general problem solving (similar to in-class work), short program segments (similar to programming assignments); Assessment of group participation on course projects, including peer-assessment of participation and contribution to the group effort.

10. **REPRESENTATIVE TEXT(S):**

Possible textbooks include:

A. Porcello & Banks. *Learning React: Modern Patterns for Developing React Apps*, 2nd ed. O'Reilly Media, 2020
B. Brown. *Web Development with Node and Express: Leveraging the JavaScript Stack*, 1st ed. O'Reilly Media, 2019
C. Dabit. *React Native in Action*, 1st ed. Manning Publications, 2019
D. Fein & Moiseev. *Angular Development with Typescript*, 1st ed. Manning Publications, 2019

Origination Date: October 2023
Curriculum Committee Approval Date: November 2023
Effective Term: Fall 2024
Course Originator: Kamran Eftekhari