UMKC Course Syllabus

COMP-SCI 497 Directed Readings

Fall 2021

Class Meeting Time & Location: Online Asynchronous

Course Description:

Full Stack Open

This course is an introduction to modern web development with JavaScript. The main focus is on single page applications implemented with React and supporting them with RESTful and GraphQL web services implemented with Node.js. The course has also parts on TypeScript, React Native and Continuous integration.

Other topics include debugging applications, configuration, managing runtime environments and NoSQL databases.

The course and all its related content is available online.

Prerequisites:

Participants are expected to have good programming skills, basic knowledge of web programming and databases, and mastery of basic use of the Git version management system. You are also expected to have perseverance and a capacity for solving problems and seeking information independently.

Previous knowledge of JavaScript or other course topics is not required.

Course Material:

The course material is meant to be read one part at a time, reading each part all the way through before moving on to the next one.

The material contains exercises, which are placed so that the preceding material provides enough information for solving each exercise. You can do the exercises as you encounter them in the material, but it can also be beneficial to read all of the material in the part before starting with the exercises.

In many parts of the course, the exercises build one larger application one small piece at a time. Some of the exercise applications are developed through multiple parts.

The course material is based on incrementally expanding example applications, which change from part to part. It's probably best to follow the code along while making small modifications independently. The code of the example applications for each step of each part can be found on GitHub.

Taking the Course:

The course contains eleven parts, the first of which is numbered 0 for historical reasons. A part corresponds loosely to one week (averaging 15-20 hours) of studying, but the speed of completing the course is flexible.

Proceeding from part *n* to part *n+1* is not sensible before good enough know-how of the topics of part *n* has been achieved. In pedagogic terms, the course uses mastery learning, and you are only intended to proceed to the next part after doing enough of the exercises of the previous part.

You are expected to do *at least* all of the exercises that are not marked with an asterisk(*).

Project:

Weeks 13-16 will focus on building a personal project of the learner's choice. The project is expected to take 60-80 hours of work (15-20 hours per week), demonstrate mastery of the topics learned in this course, and deliver *at least* a MVP. The project should not only be a web application, but it should also have an option to be a desktop application through the use of Electron.

Course Objectives:

Upon successful completion of this course, the student should be able to:

- Use JavaScript/TypeScript to build modern web applications with ReactJS, React Native, Node.js, and continuous integration.
- Utilize GraphQL and REST APIs to fulfill queries/pipelining.
- Debug, configure, and manage environments for applications.
- Design a NoSQL database to store application data.

Tentative Course Schedule

Week 1: Fundamentals of Web Apps

Week 2: Introduction to React

Week 3: Communicating with server

Week 4: Programming a server with NodeJS and Express

Week 5: Testing Express servers, user administration

Week 6: Testing React apps

Week 7: State management with Redux

Week 8: React router, custom hooks, styling app with CSS and webpack

Week 9: GraphQL

Week 10: TypeScript

Week 11: React Native

Week 12: Continuous Integration/Continuous Delivery

Week 13-16: Personal Project