Mobile App Development

Contact Information

Dan Mack: d_mack@sau9.org

Course Description

ReactJS offers graceful solutions to some of front-end programming's most persistent issues, allowing you to build dynamic and interactive web apps with ease. It's fast, scalable, flexible, powerful, and has a robust developer community that's rapidly growing. There's never been a better time to learn React. You'll develop a strong understanding of React's most essential concepts: JSX, class and function components, props, state, lifecycle methods, and hooks. You'll be able to combine these ideas in React's modular programming style. You'll begin with the front-end, move on to the back-end, then learn to connect the two. By the time you're done, you'll have the well-rounded skills needed to enter this in-demand job market.

Student Learning Outcomes

PC. 1: Students will have **programming knowledge** skills that allow students to develop solutions to novel problems.

- EU 1.1 Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).
- EU 1.2 People write programs to execute algorithms.
- EU 1.3 Programming is facilitated by appropriate abstractions.
- EU 1.4 Programs are developed, maintained, and used by people for different purposes.
- EU 1.5 A variety of abstractions built on binary sequences can be used to represent all digital data.
- EU 1.6 Multiple levels of abstraction are used to write programs or create other computational artifacts.
- EU 1.7 Computing facilitates exploration and the discovery of connections in information.
- EU 1.8 There are trade-offs when representing information as digital data.

PC. 2: Students will critically **computational skills** technical documentation and code.

- EU 2.1 Programming uses mathematical and logical concepts.
- EU 2.2 Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
- EU 2.3 Algorithms can solve many, but not all, computational problems.
- EU 2.4 Models and simulations use abstraction to generate new understanding and knowledge.
- EU 2.5 People use computer programs to process information to gain insight and knowledge.

PC. 3: Students will effectively **design and communicate** to be contributing team members

- EU 3.1 Creative development can be an essential process for creating computational artifacts.
- EU 3.2 Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
- EU 3.3 Computing can extend traditional forms of human expression and experience.
- EU 3.4 Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.
- EU 3.5 Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.

PC. 4: Students will be efficient self learners that understand **personal success skills** are critical to career success.

- EU 4.1 Computing enhances communication, interaction, and cognition.
- EU 4.2 Computing enables innovation in nearly every field.
- EU 4.3 Communication comes in a variety of forms and is necessary
- EU 4.5 Knowing your career path allows you to set goals and plan long term

PC. 5: Students will have an understanding of the **User Experience** when it comes to programming.

- EU 5.1 The Internet is a network of autonomous systems.
- EU 5.2 Characteristics of the Internet influence the systems built on it.
- EU 5.3 Cybersecurity is an important concern for the Internet and the systems built on it.

- EU 5.4 Computing has a global affect — both beneficial and harmful — on people and society.
- EU 5.5 Computing innovations influence and are influenced by the economic, social, and cultural contexts in which they are designed and used.

Course Requirements

A strong foundation in JavaScript is a prerequisite for this course, as well as basic HTML. Students should have completed Web design and AP Computer Science prior to taking this course as well.

Student Evaluation

Work Ethics	25%
Projects	35%
Challenges	25%
Exercises	15%

4 Projects

React and Redux Portfolio Project

Be introduced to web applications and learn about single page applications (SPAs) and how they are different from static websites. Be introduced to the popular JavaScript library, React. Dive further into React and learn about props, state, hooks as well as testing with Jest and Enzyme. Learn Redux, the library most commonly used with React to manage application state. Learn how to use Git and GitHub to collaborate efficiently with developers. Bring together what you have learned in the previous lessons and build a project

Competency assessed: PC.1

Back-End Portfolio Project

Start learning about back-end development and programming servers. Learn about the popular back-end environment, Node.js and how to create back-end servers and APIs in JavaScript using the popular Express.js. Learn about TDD techniques for full-stack web applications. Learn how to create tables, create, retrieve, and update data in SQL databases, and build a data-intensive web app. Much of the internet is the flow of information. Learn how databases store data so that it can be displayed on the web using PostgreSQL. Learn how to design relational databases that you can then implement in PostgreSQL. Learn about database performance and techniques for efficiently accessing data and maintaining optimal performance. Build an e-commerce REST API using Node/Express and PostgreSQL.

Competency assessed: PC.3

Full-Stack Portfolio Project

Learn about the different ways to combine your front-end and back-end systems to create a cohesive full-stack application. Learn PostgreSQL to connect JS and SQL in web apps, and build your first fully-integrated app back-end. Learn how to make a web application secure. Learn advanced Test Driven Development (TDD) concepts that will help to test your web application as a whole. Expand the e-commerce REST API with a React client app, creating a PERN (Postgres, Express, React, Node) full-stack e-commerce experience.

Competency assessed: PC.5

Final Portfolio Project

Begin interview preparation by learning the fundamentals of linear data structures. Continue interview preparation by learning advanced applications of data structures. Continue interview preparation by learning the fundamentals of algorithms. Continue interview preparation by learning the fundamentals of search algorithms. Review some key technical interview topics as well as soft skills which employers are looking for in potential candidates. Build a PERN app of your choosing and deploy it to Heroku.

Competency assessed: PC.2, PC.4

Ethical Behavior

This course’s philosophy on academic honesty is best stated as "be reasonable." This course recognizes that interactions with classmates and others can facilitate mastery of the course’s material. However, there remains a line between enlisting the help of another and submitting the work of another.

The essence of all work that you submit to the course must be your own. Collaboration on problems is not permitted (unless explicitly stated otherwise) except to the extent that you may ask classmates and others for help so long as that help does not reduce to another doing your work for you. Generally speaking, when asking for help, you may show your code or writing to others, but you may not view theirs, so long as you and they respect this policy's other constraints.

Late work

Assignments are due on the announced due date. It is your responsibility to keep up with class activities and assignments and request missing assignments due to absence. Upon returning from an excused absence, students will be given two days for each day absence to make up missed assignments. All students work at a different pace and will be graded primarily on their quality of work and productivity level during class. As long as the students are highly productive each day and producing work of high quality, they will receive excellent grades. Extensions will also be available upon request.

Food/Drink

Food and drinks are not permitted near the computer stations. There will be dedicated stations for water bottles or snacking.

Electronic Devices

Phones and other electronic devices are allowed in class if they do not become a distraction (texting, playing games, checking social media, web browsing, etc.). Students who are regularly off task or behind in their work, will have their phone privileges revoked. However, during instructional time, tests and quizzes, electronic devices are not to be used at all (unless directed to by the teacher). The CTC has a general no cellphones during instruction time policy. This means your phone should not be visible during lecture, going over examples, group work. You will get one warning to put the phone away. If we are working on individual work you are welcome to ask to use your phone, if you don't ask permission you will get a warning. After the warning your phone will go into one of the phone jails and you can collect it at the end of the class.

Computer Use

The classroom computers and related devices are to be used for classwork only. Do not download any files or programs not related to your classwork. Do not change the Login screen background. Do not install any program without permission of the teacher. Do not run any unapproved programs (Minecraft, Call of Duty, Halo, etc.), even from a network, external drive or remote device. Do not view or download any images, videos, or sound files that are offensive, racist, promote violence or drug use, etc.

Outline

- **React and Redux Portfolio Project**
    - **Web Apps**: Be introduced to web applications and learn about single page applications (SPAs) and how they are different from static websites.
    - **React, Part I**: Be introduced to the popular JavaScript library, React.
    - **React, Part II**: Dive further into React and learn about props, state, hooks as well as testing with Jest and Enzyme.
    - **Redux**: Learn Redux, the library most commonly used with React to manage application state
    - **Git and GitHub**: Learn how to use Git and GitHub to collaborate efficiently with developers.

- **Back-End Portfolio Project**
    - **Basics of Back-End Development**: Start learning about back-end development and programming servers.
    - **Build a Back-End with Node/Express.js**: Learn about the popular back-end environment, Node.js and how to create back-end servers and APIs in JavaScript using the popular Express.js
    - **Back-End and Feature Testing**: Learn about TDD techniques for full-stack web applications.
    - **SQL for Back-End Development**: Learn how to create tables, create, retrieve, and update data in SQL databases, and build a data-intensive web app.
    - **PostgreSQL Database**: Much of the internet is the flow of information. Learn how databases store data so that it can be displayed on the web using PostgreSQL.
    - **Designing Relational Databases**: Learn how to design relational databases that you can then implement in PostgreSQL
    - **Advanced PostgreSQL**: Learn about database performance and techniques for efficiently accessing data and maintaining optimal performance.

- **Full-Stack Portfolio Project**
    - **Connecting Front-End to Back-End**: Learn about the different ways to combine your front-end and back-end systems to create a cohesive full-stack application.
    - **Adding a PostgreSQL Database**: Learn PostgreSQL to connect JS and SQL in web apps, and build your first fully-integrated app back-end
    - **Security, Authentication, and Authorization**: Learn how to make a web

application secure.

  - **Advanced Concepts in TDD**: Learn advanced Test Driven Development (TDD) concepts that will help to test your web application as a whole.

- **Final Portfolio Project**
  - **Linear Data Structures**: Begin interview preparation by learning the fundamentals of linear data structures.
  - **Complex Data Structures**: Continue interview preparation by learning advanced applications of data structures.
  - **Algorithms**: Continue interview preparation by learning the fundamentals of algorithms.
  - **Search & Graph Search Algorithms**: Continue interview preparation by learning the fundamentals of search algorithms.
  - **Interview Skills**: Review some key technical interview topics as well as soft skills which employers are looking for in potential candidates.