

Web Programming

1. Course number and name: 020PWBES3 Web Programming

2. Credits and contact hours: 4 ECTS credits, 2x1:15 contact hours

3. Instructor's or course coordinator's name: Anthony Tannoury

4. Instructional materials:

Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js. Shama Hoque. Packt Publishing Ltd, 2020.

a. Other supplemental materials:

- Full Stack Web Development with React Specialization. Jogesh K. Muppala. Massive Online Open Courses (MOOCs) on Coursera.
- Full Stack Web Development with Angular Specialization. Jogesh K. Muppala. Massive Online Open Courses (MOOCs) on Coursera.
- Angular for Enterprise-Ready Web Applications. Doguhan Uluca. Packt Publishing Ltd, 2020.
- Learning React: functional web development with React and Redux. Alex Banks and Eve Porcello. O'Reilly Media, Inc., 2017.
- React's official documentation. Facebook Open Source. Web.

5. Specific course information

a. Catalog description:

This course covers the development of web applications on both the front-end (client-side) and the back-end (server-side). It is, in fact, a hands-on web programming course where a MongoDB, Express, React and Node (MERN) web application is gradually designed and implemented as the course progresses.

The course first introduces the basic languages used for web development, namely HTML, CSS and JavaScript. They are followed by the introduction of the Twitter Bootstrap web framework and the quick implementation of several web pages using this framework. Afterwards, the React framework along with its underlying Flux architecture is explained. A React Single Page Application (SPA) is then implemented. At this stage, the front-end has been fully implemented while the back-end is still mocked using a simulated JSON-Server. This mock backend is then replaced by a fully functional REST API implemented using Node.js, the Express framework and the MongoDB database. This REST API is then tested using Postman before it is integrated with the react front-end, concluding the implementation of a full stack MERN web application.

Each part of this full stack MERN application can now be deployed on a cloud provider such as Heroku to provide Software as a Service (SaaS) functionalities. We then introduce Google Firebase which provides Backend as a Service (BaaS) functionalities to discharge the developer from implementing a backend. We then conclude with an initiation to Angular as a possible alternative to React for building enterprise full stack MongoDB, Express, Angular and Node (MEAN) web applications.

b. Prerequisites: None

c. Selected Elective for CCE and EE students

6. Educational objective for the course

a. Specific outcomes of instruction:

- Implement an interactive web page using HTML, CSS and JavaScript.
- Design and implement a user-friendly web interface using the Twitter Bootstrap framework.
- Design and implement a modular web application front-end using the React framework. Test this front with a mocked JSON-server backend.
- Conceive and construct a JSON document database using MongoDB. Access this database using MongoDB Compass.
- Design and implement, for the web application's backend, a REST API using Node.js, the Express framework and MongoDB. Test this REST API using Postman.
- Write high quality code in accordance with the best practices of Web development.
- Implement a Software as a Service (SaaS) deployed on Heroku.
- Explain the main components of Angular and Google Firebase.

b. PI addressed by the course:

PI	1.3	2.1	2.3	2.4	2.5	3.2	5.2	7.2
Covered	x	x	x	x	x	x	x	x
Assessed	x	x	x			x		

7. Brief list of topics to be covered

- Course introduction. HTML & CSS. (1 lecture)
- The JavaScript programming language and the Document Object Model (DOM) (2 lectures)
- The Twitter Bootstrap UI framework: the grid system and responsive web design, the navigation bar, forms and buttons, badges, labels, alerts, progress bars, images, media objects, thumbnails, tables, panels, tabs, wells, pills, accordion, collapse, scrollspy, affix, modals, tooltips, popovers, carousel (3 lectures)
- Introduction to React : JSX, functional components, hooks, class components, state, props, controlled forms, uncontrolled forms, react animations, react router et Single Page Applications (SPAs). (4 lectures)
- React and Redux : the Flux Architecture, the redux store, redux forms, redux actions, combining reducers, redux thunk or redux saga. « Representational State Transfer (REST) » APIs. Fetching data using the Fetch API. Debugging the redux store. Testing the front-end by mocking the backing through the use of JSON-server. (4 lectures)
- The Node.js JavaScript runtime: the npm package manager, Node modules and the efficient use of the non-blocking IO model (1 lecture)

- The Express web framework for Node.js: building a server-side REST API, the Express router and the Express Generator. Testing the backend using Postman and iteratively improve it. (2 lectures)
- Document databases: MongoDB and the Mongoose Object-Document Mapper (ODM), Mongoose Population and relationships between documents. Introduction to conceiving a MongoDB database schema. Accessing the database using MongoDB Compass. (3 lectures)
- Backend security : user passport authentication. OAuth for Facebook or Google authentication. Certificate Generation and HTTPS. Test the backend using Postman and iteratively improve it. (2 lectures)
- Finalizing the backend: File Upload et CORS. Test the final backend using Postman. (2 lectures)
- Full MERN application: connect the React front-end React the API REST implemented by the backend using Node.js, Express et MongoDB. Deploy both the frontend and backend on Heroku to enable Software as a Service (SaaS) functionalities. The possible use of a Backend as a Service (BaaS) like Firebase as an alternative to costly backend developments. (2 lectures)
- Introduction to the Angular framework: « Model View Controller (MVC) », modules, controllers, scope, forms, directives, filters, factory, service, dependency injection, templates, routing, Single Page Applications (SPAs). (2 lectures)