**SWE-452: Enterprise Application Development (EAD)**

**General Information**

|  |  |
| --- | --- |
| **Course Number** | SWE-452 |
| **Credit Hours** | 3+0 (Theory Credit Hour = 3, Lab Credit Hours = 0) |
| **Prerequisite** |  |
| **Semester** | VII |

## Course Objectives / Description

This course is aimed to teaches students how to create robust enterprise applications using latest web technologies that allow for rapid growth and change. Advanced application development for enterprise level computing involves the development of the mid to large applications. In this course, emphasis is placed on version control system, web services, performance, security, front-end & back-end development, testing & application deployment. By the end of the course, students will have practical experience of the different tools and technologies for the enterprise level application.

## Course Learning Outcomes (CLOs)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Course Learning Outcome | Domain | Level | Assessment Tool |
| C1 | To understand the various concepts of the enterprise applications | C | 2 | LMS |
| C2 | To develop web application by developing and consuming web services | C | 3 | LMS |
| C3 | Test and deploy application | C | 4 | LMS |

Domains: C=Cognitive, A=Affective, P=Psychomotor

Levels:

Cognitive = {1: Remembering, 2: Understanding, 3: Applying, 4: Analyzing, 5: Evaluating, 5: Creating}

Affective = {1: Receiving, 2: Responding, 3: Valuing, 4: Organizing, 5: Characterizing}

Psychomotor= {1: Imitation, 2: Manipulation, 3: Precision, 4: Articulation, 5: Naturalization}

**Course Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week No.** | **Topic** | **Suggested Readings (Chapters)** | **CLO** |
| 1 & 2 | * Introduction of Enterprise Application Development * Working with Git & GitHub   + Git clone   + Git add   + Git commit   + Git status   + Git push   + Git pull   + Merge Conflicts   + Git log   + Git reset   + Feature branching   + Git fetch   + Fork   + Pull request   + Deploy a repository on Github.io | Teacher Notes | C1 |
| 3 & 4 | * Introduction to ReactJS * Core features of ReactJS * Babel * Create with npx create-react-app & Vite * Virtual DOM * JSX * Rendering Elements * Components & Props   + Function and Class Components   + Rendering a component   + Composing a component   + Extracting components   + Props are read-only * State * Handling Events * Conditional Rendering * Lifecycle methods * Using Formik to handle forms   + Simple Form   + useFormik Hook   + Managing Form Sate   + Handling Form Submission   + Form Validation   + Display Error Messages   + Visited Fields   + Schema Validation with Yup | Teacher Notes | C1 |
| 5 & 6 | * React Router Introduction   + Install & Setup   + Configuring Routes   + Links & Active Links   + Navigating Programmatically   + Index Route   + Nested Route   + No Match Route   + Dynamic Route   + Outlet   + URL Params   + Search Params   + Create Search Params   + Absolute & Relative Links | Book  &  Teacher Notes | C1 & C2 |
| 7 | * Hooks   + State Hook   + Effect Hook   + Context Hook   + Reducer Hook   + Ref Hook   + useReducer with useContext   + Building your own Hooks * Higher-Order Components * Render Props * Context API * Context API + useReducer | Book  &  Teacher Notes | C1 & C2 |
| **First Mid Exam** | | | |
| 9 | * State Management using Redux   + Core concepts   + Immutable state tree   + State changes with actions   + Pure & Impure functions   + Reducer function   + Store | Book  &  Teacher Notes | C1 & C2 |
| 10 | NodeJS   * Getting Started with NodeJS   + Introduction   + Runtime Architecture   + Understanding event-driven architecture   + Node.js vs Traditional server-side scripting   + Setting up Environment   + File & directory structure   + Module Types   + Working with File System   + A simple server with Node   + Serving static resources   + Working with Nodemon | Book  &  Teacher Notes | C1 & C2 |
| 11-12 | * Introduction to ExpressJS   + Views and Layouts   + Routing   + Middleware     - Common     - Third-party   + Handling Requests & Response   + Static Files and Views   + Dynamic Content in Views   + Building RESTful APIs * Template Engines   + EJS | Book  &  Teacher Notes | C1, C2  &  C3 |
| 13 | * Working with MongoDB   + Introduction   + Collections and Documents   + Creating, Listing & using Databases   + ORM vs ODM   + Using MongoDB with Mongoose (ODM)   + CRUD Operations | Book  &  Teacher Notes | C1 & C2 |
| 14 | * Form Handling   + Sending client data to Server   + HTML Forms   + Form Handling with Express   + File Uploads * Cookies & Sessions | Book  &  Teacher Notes | C1, C2  &  C3 |
| 15-16 | * Socket.io * Authentication and Security   + Authentication vs Authorization   + Using JSON Web Token JWT   + Password hashing with bcrypt   + OAuth implementation (Google, Facebook) * CORS and Rate Limiter * Unit Testing   + JEST   + Enzyme * Debugging * Deployment | Book  &  Teacher Notes | C1, C2  &  C3 |
| **Final Exam** | | | |

**CLO-PLO Map**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Graduate Attribute (PLOs)** | | | | | | | | | | | | |
| **CLOs** | **GA1** | **GA2** | **GA3** | **GA4** | **GA5** | **GA6** | **GA7** | **GA8** | **GA9** | **GA10** | **GA11** | **GA12** |
| CLO 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CLO 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| CLO 3 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Textbook**

1. Web Development with Node and Express Leveraging the JavaScript Stack-O'Reilly Media by Ethan Brown - (2019)
2. The Road to learn React: Your journey to master plain yet pragmatic React.js by Robin Wieruch
3. Fullstack React The Complete Guide to ReactJS and Friends by Accomazzo Anthony, Murray Nathaniel, Lerner Ari (2017)

**Reference Material**

Available on LMS

**Instructor**

|  |  |
| --- | --- |
| **Name** | Khalid Hussain |
| **Designation** | Lecturer |
| **Department** | Computer Science |

**Computer Science/Software Engineering**

**Program Learning Outcomes**

**GA: Graduate Attributes**

**GA1 Computing Knowledge:** An ability to apply knowledge of mathematics, science, computing fundamentals and computing specialization to the solution of complex computing problems.

**GA2 Problem Analysis:** An ability to identify, formulate, research literature, and analyze complex computing problems reaching substantiated conclusions using first principles of mathematics, natural sciences and computing sciences.

**GA3 Design/Development of Solutions:** An ability to design solutions for complex computing problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

**GA4 Investigation:** An ability to investigate complex computing problems in a methodical way including literature survey, design and conduct of experiments, analysis and interpretation of experimental data, and synthesis of information to derive valid conclusions.

**GA5 Modern Tool Usage:** An ability to create, select and apply appropriate techniques, resources, and modern IT tools, including prediction and modeling, to complex computing activities, with an understanding of the limitations.

**GA6 The Computer Scientist and Society:** An ability to apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional computing practice and solution to complex computing problems.

**GA7 Environment and Sustainability:** An ability to understand the impact of professional computing solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

**GA8 Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of computing practice.

**GA9 Individual and Teamwork:** An ability to work effectively, as an individual or in a team, on multifaceted and /or multidisciplinary settings.

**GA10 Communication:** An ability to communicate effectively, orally as well as in writing, on complex computing activities with the computing community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**GA11 Project Management:** An ability to demonstrate management skills and apply computing principles to one’s own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.

**GA12 Lifelong Learning:** An ability to recognize importance of, and pursue lifelong learning in the broader context of innovation and technological developments