| 24UETES101 | Application Development Practices | 3/0/2 |
|---|---|---|

| | |
|---|---|
| **Nature of Course:** | Theory & Programming |
| **Prerequisites:** | NIL |

## Course Objectives:

| | |
|---|---|
| 1 | To learn various software development models. |
| 2 | To learn extreme programming and use of GITHUB to manage collaborative software development . |
| 3 | To Learn simple web page development using HTML. |
| 4 | To design interactive web page development using CSS. |
| 5 | To understand bootstrap process. |

## Course Outcomes:

**Upon completion of the course, students shall have ability to**

| | | |
|---|---|---|
| 1 | Understand fundamental software development concepts and tools. | [U] |
| 2 | Choose methodologies and tools to develop and deploy applications. | [AP] |
| 3 | Apply HTML knowledge for static web page development. | [AP] |
| 4 | Apply CSS knowledge for designing and implementing dynamic web page. | [AP] |
| 5 | Implement web site using java script . | [AP] |

## CO-PO Mapping:

Mapping of Course Outcomes to Program Outcomes (POs) & Program Specific Outcomes (PSOs):

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 2 | 1 | 1 | - | - | - | - | - | - | - | 2 | 3 | 2 | 2 |
| CO2 | 3 | 2 | 3 | 3 | 3 | 2 | - | - | - | - | - | 2 | 3 | 3 | 2 |
| CO3 | 3 | 2 | 3 | 3 | 3 | 2 | - | - | - | - | - | 2 | 3 | 3 | 2 |
| CO4 | 3 | 2 | 3 | 3 | 3 | 2 | - | - | - | - | - | 2 | 3 | 3 | 2 |
| CO5 | 3 | 2 | 3 | 3 | 3 | 2 | - | - | - | - | - | 1 | 3 | 3 | 2 |

## Teaching-Learning & Assessment Scheme

| Learning Scheme | | | Credits | Assessment Scheme | | | Summative Assessment | Total |
|---|---|---|---|---|---|---|---|---|
| | | | | Formative Assessment | | | | |
| L | T | P | | CIA TH1 | CIA TH2 | Capstone model Assignment | End Semester Exam | |
| 3 | 0 | 0 | 3 | 25 | 25 | 20 | 60 Scaled Down 30 | 100 |

## Course Contents:

**Unit I Software Development Model** **9 Hrs**

Introduction to software development process, Software engineering principles, requirements gathering and analysis, Overview of programming languages. Agile Software Development: Traditional Software Development Models, Agile methodology (Scrum, Kanban), DevOps practices, Comparison of methodologies.

**Case Study:**

Develop a simple application that can perform basic operations using the chosen programming language.

**Unit II Extreme Programming and GitHub** **9 Hrs**

The Rules of Extreme Programming, Extreme Programming (XP), Principles, Extreme Programming (XP) – Key Terms, Introduction to Lean Software Development, Principles of Lean Software Development, What is Kanban? Introduction to Git, Getting a Git Repository, Recording Changes to the Repository, Viewing the Commit History, Undoing Things, Working with Remotes, Tagging, Git Aliases, Git Branching, Branches in a Nutshell, Basic Branching and Merging, Branch Management, Remote Branches, Rebasing. Introduction to GitHub, Introduction, Set up Git, Create a repository, GitHub Flow, Contribution to Projects, Communicating on GitHub. Linux Basic CommandsX.

**Case Study**:

Imagine a scenario where a team of developers is working on a mobile application. Students will simulate this scenario by collaborating on a shared Git repository to develop a simple mobile app feature. They will create branches for different features, collaborate on code changes, resolve conflicts, and merge their changes back into the main branch. This case study will demonstrate the importance of version control in collaborative software development.

**Unit III HTML** **9 Hrs**

Understand the structure of an HTML page, New Semantic Elements in HTML 5, Learn to apply physical/logical character effects, Learn to manage document spacing. Tables, Understand the structure of an HTML table, Learn to control table format like cell spanning, cell spacing, border. List, Numbered List, Bulleted List, Working with Links, Understand the working of hyperlinks in web pages, Learn to create hyperlinks in web pages, Add hyperlinks to list items and table contents. Image Handling, Understand the role of images in web pages, Learn to add images to web pages, Learn to use images as hyperlinks.

**Case Study:**

Building an Engaging Website with HTML Fundamentals.

**Unit IV Introduction to Cascading Style Sheets** **9 Hrs**

What CSS can do, CSS Syntax, Types of CSS. Working with Text and Fonts - Text Formatting, Text Effects, Fonts. CSS Selectors - Type Selector, Universal Selector, ID Selector, Class selector. Colors and Borders, Background, Multiple Background, Colors RGB and RGBA, HSL and HSLA, Borders, Rounded

Corners, Applying Shadows in border, Implementing CSS3 in the "Real World", Modernizr, HTML5 Shims, SASS, and Other CSS Preprocessors, CSS Grid Systems, CSS Frameworks

**Case Study:**

Consider a scenario where a software company is tasked with maintaining an existing codebases that lacks proper documentation and has accumulated technical debt over time. Students will analyze the codebase, identify areas for improvement, and refactor the code to adhere to clean code principles. They will then write unit tests to ensure that the refactored code behaves as expected, demonstrating the importance of clean code and testing in software maintenance.

**Unit V Bootstrap** **9 Hrs**

Introduction, Getting Started with Bootstrap, Bootstrap Basics, Bootstrap grid system, Bootstrap Basic Components, Bootstrap Components, Page Header, Breadcrumb, Button Groups, Dropdown, Nav & Navbars. JavaScript Essentials, Var, Let and Const keyword, Arrow functions, default arguments, Template Strings, String methods, Object de-structuring, Create, apply, prototype, bind method, Spread and Rest operator, Typescript Fundamentals, Types & type assertions, Creating custom object types, function types, Typescript OOPS, Classes, Interfaces, Constructor, Decorator & Spread Operators Race.

**Case Study**:

Imagine a scenario where a team of developers is working on a web application that needs to be deployed to a cloud platform. Students will set up a CI/CD pipeline using tools like Jenkins or GitHub Actions to automate the build, test, and deployment process. They will deploy the application to a cloud platform (e.g., AWS, Heroku) and configure the pipeline to trigger automatic deployments whenever new changes are pushed to the repository. This case study will showcase the benefits of CI/CD in streamlining the software development lifecycle.

| | |
|---|---|
| **Total Hours:** | **45** |

## Laboratory Component:

| S. No | List of Experiments |
|---|---|
| **1** | Gather application specific requirements for assimilate into RE (Requirements engineering) model. |
| **2** | Select relevant process model to define activities and related tasks set for assigned project. |
| **3** | Set up a Git repository and perform basic version control operations (commit, push, pull, merge). |
| **4** | Collaboratively work on a project using Git, branching, and merging. |
| **5** | Perform manual testing of a small software application and document test cases and results. |
| **6** | Write unit tests for a simple function or module using a testing framework (e.g., JUnit for Java, unittest for Python). |
| **7** | Refactor a piece of code to improve its readability, following clean code principles. |
| **8** | Work on an agile project using Scrum or Kanban, with iterative development sprints. |
| **9** | Set up a CI/CD pipeline using a tool like Jenkins or GitHub Actions for a sample application. |

| | |
|---|---|
| **10** | Deploy a web application to a cloud platform (e.g., AWS, Heroku) using CI/CD pipelines. |
| **Project Work (Capstone)** | |
| | Collaboratively develop a small software application using version control, testing, and deployment practices learned during the course.<br>Present the project to the class, demonstrating key features, code quality, and deployment process. |
| **Text Books:** | |
| 1. | Software Engineering: A Practitioner's Approach" by Roger S. Pressman (2023). |
| 2. | Jeff Sutherland, “Scrum the Art of Doing Twice the Work in Half the Time”, Random House Publisher, 1st Edition, (2022). |
| 3. | Scott Chacon, Ben Straub, “Pro GIT”, Apress Publisher, 3rd Edition, (2024). |
| 4. | Code Complete: A Practical Handbook of Software Construction" by Steve McConnell (2014). |
| 5. | Refactoring: Improving the Design of Existing Code" by Martin Fowler (2018). |
| 6. | The Art of Software Testing" by Glenford J. Myers, Corey Sandler, and Tom Badgett (2011). |
| 7. | Continuous Integration: Improving Software Quality and Reducing Risk" by Paul M. Duvall, Steve Matyas, and Andrew Glover (2007). |
| **Reference Books:** | |
| 1. | Robert C. Martin, “Agile Software Development, Principles, Patterns and Practices”, Prentice Hall, 2nd Edition, (2020). |
| 2. | Mike Cohn, “User Stories Applied: For Agile Software”, Addison Wesley, 2nd Edition, (2023). |
| 3. | Software Engineering: A Practitioner's Approach" by Roger S. Pressman (2023). |
| 4. | Scrum: The Art of Doing Twice the Work in Half the Time" by Jeff Sutherland (2023). |
| 5. | Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley (2023). |
| 6. | Pro Git" by Scott Chacon and Ben Straub (2023). |
| 7. | Version Control with Git" by Jon Loeliger and Matthew McCullough (2020). |
| 8. | Jenkins: The Definitive Guide" by John Ferguson Smart (2020). |
| 9. | The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford (2023). |
| **Web References:** | |
| 1. | https://www.coursera.org/specializations/agile-development |
| 2. | https://www.edx.org/learn/agile |
| 3. | https://nptel.ac.in/courses/106/105/106105182/ |

| | |
|---|---|
| 4. | https://www.agilealliance.org/ |
| 5. | https://www.scrum.org/ |
| 6. | https://trello.com/ |
| **Online Resources:** | |
| 1. | http://www.agilenutshell.com/ |
| 2. | https://www.atlassian.com/agile/scrum |
| 3. | https://www.youtube.com/user/AgileMikeCohn |
| 4. | https://guides.github.com/ |
| 5. | https://git-scm.com/doc |

| | | |
|---|---|---|
| **Course Incharge** | **Course Coordinator** | **Program Coordinator** |
| **( Souhardya Gayen )** | **( Aparupa Chakraborty )** | **( Dr. K. Vengatesan )** |