HTML

Course Title: HTML Fundamentals

Course Description:

This course covers the fundamentals of HTML, the markup language used for creating web pages. Students will learn how to structure content using HTML tags, add links, images, and multimedia, and use forms for user input. The course also covers accessibility best practices and responsive design.

Learning Objectives:

By the end of this course, students will be able to:

1. Create well-structured web pages using HTML tags and attributes.
2. Add links, images, and multimedia content to web pages.
3. Use forms to collect user input and validate form data.
4. Apply accessibility best practices to ensure web content is accessible to all users.
5. Implement responsive design techniques to optimize web content for different screen sizes.

Course Outline:

Week 1:

- Introduction to HTML
- HTML Document Structure and Syntax
- Head and Meta Tags

Week 2:

- Text Formatting and Semantic Tags
- Links and Anchors
- Images and Multimedia

Week 3:

- Tables and Lists
- Forms and User Input
- Validating Form Data

Week 4:

- Accessibility Best Practices
- CSS Basics for Styling HTML
- Responsive Design Techniques

Week 5:

- HTML5 Features and APIs
- Local Storage and Session Storage
- Web Workers and Service Workers

Week 6:

- Final Project and Presentations

Assessment:

- Weekly assignments and quizzes
- Mid-term exam
- Final project and presentation

Prerequisites:

- Basic understanding of computer and internet usage
- No prior programming knowledge required.

CSS

Course Title: CSS Fundamentals

Course Description:

This course covers the fundamentals of CSS, the styling language used for creating web pages. Students will learn how to apply CSS styles to HTML elements, use CSS selectors to target specific elements, and apply CSS layout techniques for creating responsive web pages. The course also covers advanced topics such as animations, transformations, and Flexbox/Grid layout.

Learning Objectives:

By the end of this course, students will be able to:

1. Apply CSS styles to HTML elements using inline, internal, and external stylesheets.
2. Use CSS selectors to target specific elements and apply styles to them.
3. Create CSS layouts using floats, positioning, and Flexbox/Grid layout.
4. Implement CSS animations, transformations, and transitions.
5. Optimize CSS code for performance and maintainability.

Course Outline:

Week 1:

- Introduction to CSS
- CSS Syntax and Selectors
- Box Model and Layout

Week 2:

- Typography and Text Styling
- Colors, Backgrounds, and Gradients
- Responsive Design with Media Queries

Week 3:

- CSS Frameworks and Preprocessors
- CSS Resets and Normalization
- CSS Optimization and Performance

Week 4:

- Animations and Transitions
- Transformations and Transforms
- Advanced CSS Layout with Flexbox and Grid

Week 5:

- CSS Best Practices and Naming Conventions
- CSS3 Features and Browser Compatibility
- Cross-Browser Compatibility and Testing

Week 6:

- Final Project and Presentations

Assessment:

- Weekly assignments and quizzes
- Mid-term exam
- Final project and presentation

Prerequisites:

- Basic understanding of HTML and web page structure
- No prior programming knowledge required, but some familiarity with programming concepts is recommended.

•

JavaScript

Course Title: JavaScript Fundamentals
Course Description:
This course covers the basics of JavaScript programming language, including syntax, data types, control structures, functions, and objects. Students will learn how to create interactive web pages using JavaScript and DOM manipulation. The course also covers asynchronous programming, error handling, and debugging techniques.
Learning Objectives:
By the end of this course, students will be able to:
1. Write basic JavaScript programs using variables, data types, and control structures.
2. Create and use functions and objects in JavaScript programs.
3. Manipulate the Document Object Model (DOM) to create dynamic web pages.
4. Implement asynchronous programming techniques using callbacks, promises, and async/await.
5. Debug JavaScript code and handle errors effectively.

Course Outline:
Week 1:
- Introduction to JavaScript
- Variables and Data Types
- Control Structures

Week 2:
- Functions and Objects
- Arrays and Loops

Week 3:
- Document Object Model (DOM)
- Events and Event Handlers

Week 4:
- Asynchronous Programming with Callbacks
- Promises and Async/Await

Week 5:
- Error Handling and Debugging
- Best Practices and Code Organization

Week 6:
- Final Project and Presentations

Assessment:
- Weekly assignments and quizzes
- Mid-term exam
- Final project and presentation

Prerequisites:
- Basic understanding of HTML and CSS
- Familiarity with a programming language (e.g. Python, Java, C++) is recommended, but not require
•

React js

Course Title: React Fundamentals

Course Description:

This course covers the fundamentals of React, a popular JavaScript library for building user interfaces. Students will learn how to create reusable components, manage state and props, and use React hooks for managing state in functional components. The course also covers React Router for handling client-side routing and Redux for state management.

Learning Objectives:

By the end of this course, students will be able to:

1. Build and test React components using JSX syntax.
2. Manage state and props in React components.
3. Use React hooks to manage state in functional components.
4. Implement client-side routing using React Router.
5. Manage application state using Redux.

Course Outline:

Week 1:

- Introduction to React
- JSX and Component Rendering
- Props and State

Week 2:

- Component Lifecycle Methods
- Event Handling and Forms
- Composition and Reusability

Week 3:

- React Hooks
- Higher-Order Components
- Testing React Components

Week 4:

- React Router
- Server-side Rendering
- Performance Optimization

Week 5:

- Redux Basics
- Reducers and Actions
- Async Actions and Middleware

Week 6:

- Final Project and Presentations

Assessment:

- Weekly assignments and quizzes
- Mid-term exam
- Final project and presentation

Prerequisites:

- Basic understanding of HTML, CSS, and JavaScript

- Familiarity with a programming language (e.g. Python, Java, C++) is recommended, but not required.

GIT

Course Description:
This course covers the fundamentals of Git, a popular version control system used for software development. Students will learn how to use Git for managing source code, collaborate with other developers using Git, and contribute to open-source projects on GitHub. The course also covers advanced Git topics such as branching and merging, resolving conflicts, and rebasing.
Learning Objectives:
By the end of this course, students will be able to:
1. Create and manage Git repositories for source code management.
2. Collaborate with other developers using Git.
3. Contribute to open-source projects on GitHub.
4. Use Git branching and merging for parallel development.
5. Resolve conflicts and use Git rebase to keep history clean.

Course Outline:
Week 1:
- Introduction to Version Control Systems
- Git Installation and Configuration
- Basic Git Commands and Workflow

Week 2:
- Branching and Merging in Git
- Collaborating with Git using GitHub
- Pull Requests and Code Reviews

Week 3:
- Git Stash and Rebase
- Resolving Conflicts in Git
- Git Hooks and Customizations

Week 4:
- Git Workflows and Best Practices
- Git Performance and Optimization
- Git for Open-Source Contributions

Week 5:
- Advanced Git Topics: Submodules, Cherry-Picking, and Bisect
- Git GUI Tools and Clients
- Git and Continuous Integration/Deployment (CI/CD)

Week 6:
- Final Project and Presentations

Assessment:
- Weekly assignments and quizzes
- Mid-term exam
- Final project and presentation

Prerequisites:
- Basic understanding of programming concepts and command-line usage

- Familiarity with a programming language (e.g. Python, Java, C++) is recommended, but not required.

Data Structures and Algorithms

Course Title: Data Structures and Algorithms
Course Description:
This course covers the fundamentals of data structures and algorithms used in computer programming. Students will learn how to implement and analyze various data structures such as arrays, linked lists, stacks, queues, trees, and graphs. The course also covers algorithms for searching, sorting, and graph traversal.
Learning Objectives:
By the end of this course, students will be able to:
1. Implement and use basic data structures such as arrays, linked lists, stacks, and queues.
2. Implement and use advanced data structures such as trees, heaps, hash tables, and graphs.
3. Analyze the time and space complexity of algorithms and data structures.
4. Implement and use common searching and sorting algorithms.
5. Implement and use graph algorithms for traversal and shortest path.

Course Outline:
Week 1:
- Introduction to Data Structures and Algorithms
- Analysis of Algorithms and Big O Notation
- Arrays and Dynamic Arrays

Week 2:
- Linked Lists
- Stacks and Queues
- Recursion and Backtracking

Week 3:
- Trees and Binary Trees
- Heaps and Priority Queues
- Hash Tables

Week 4:
- Graphs and Graph Representations
- Graph Traversal Algorithms (BFS and DFS)
- Shortest Path Algorithms (Dijkstra and Bellman-Ford)

Week 5:
- Searching Algorithms (Linear Search, Binary Search)
- Sorting Algorithms (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort)
- Dynamic Programming

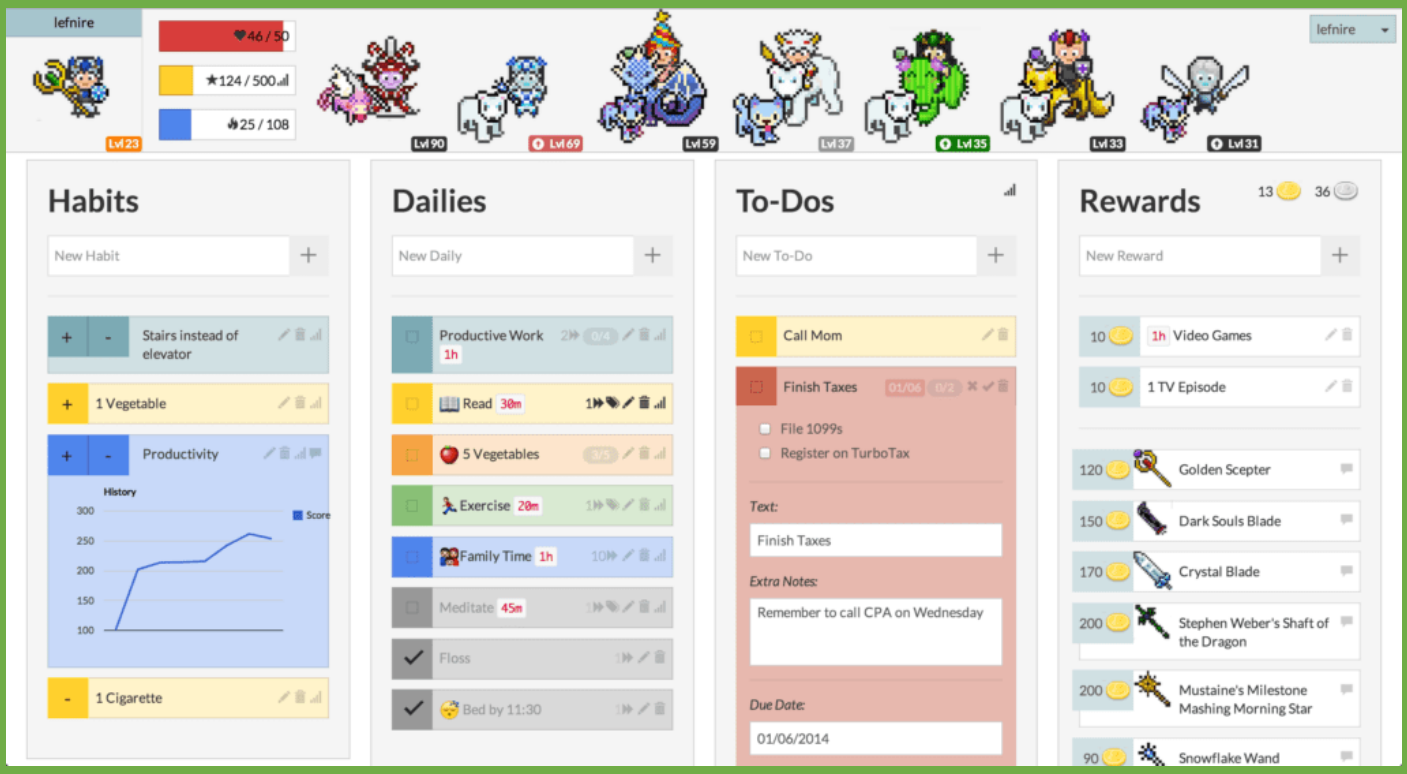
Week 6:
- Final Project and Presentations

Assessment:
- Weekly assignments and quizzes
- Mid-term exam
- Final project and presentation

Prerequisites:

- Basic understanding of programming concepts and a programming language (e.g. Python, Java, C++)

PROJECT



RESUME PREPRATION

PIRATE KING

LinkedIn | 123-456-7890 | piratekingdom.com | pirateking@gmail.com | GitHub

Skills

- C# | .NET | Java | JavaScript | TypeScript | C++ | C | CosmosDB | MSSQL | Node | Express | React | Vue | Redux | jQuery | NoSQL | Git
- Azure | Cloud Computing | CI/CD | XUnit | Jest | Cucumber | Nightwatch | Unit Testing | Lambda | OOP | Unity 2D | Game Development
- Microservices | Distributed Systems | Frontend | Backend | Full-Stack | English, Korean, Japanese – *All professional proficiency or above*

Experience

YouTuber — YouTube — **07/2021 - Current**

- **PIRATE KING**, Software Engineering, SWE Skits & Entertainment, Tech Life, Coding, Career Advice, LeetCode

Software Engineer — **Microsoft** — *Redmond, WA, USA* — **05/2018 - 04/2022**

- Led the design and development of multiple enterprise-level microservice applications of Commerce Experience Group, driving $35.3 billion of revenue every year using the latest technologies of **Azure, C#, .NET, Cosmos DB, Azure Functions, Key Vault, and MS Graph.**
- Designed and implemented scalable APIs and background workers for managing first- and third-party proprietary licenses using **.net Core, Azure Functions**, and other **Azure cloud technologies** that serve millions of license requests daily.
- Led the development of several products E2E, from identifying system requirements and partner dependencies to workload balancing, software implementation, engineering, testing, and configuring metrics, alarms, monitors, and dashboards.
- Enriched system metrics by integrating the platforms with telemetry; facilitated in-depth logging by correlating APIs with vector contexts.
- Continuous Integration/Deployment Pipeline Integration, pull requests, code reviews, load/stress testing, unit/integration/e2e testing

Software Development Engineer — **Amazon** — *Seattle, WA, USA* — **04/2017 - 04/2018**

- Implemented enterprise applications of Prime's Content Experiment Platforms using **Java, React**, AngularJS, AWS, and DynamoDB.
- Designed and developed systems facilitating marketers to perform various optimization experiments within the Prime ecosphere.
- Automated and optimized business logic for the core marketing experiments, including A/B, Auto-Targeting, and Multivariate Testing.
- Completely automated the marketing platforms' user-experience testing process by integrating Nightwatch Selenium.

Software Engineer — **ebay** — *Seoul, South Korea* — **12/2014 - 03/2017**

- Designed and implemented enterprise fintech applications of South Korea's largest e-commerce platforms (Gmarket, Auction, SmilePay), driving monthly revenue of $1 billion using **C# .NET, MVC, MSSQL**, node, react, redux, and jQuery.
- ***Reported directly to CPO:*** Engineering lead for designing and developing the fintech transaction dashboard that provides a rich visual summary of daily user purchase patterns powered by Google Chart. Used by the board of directors in the decision-making process.
- Engineering owner of the E2E experience of the commerce platform's cancel, return, and exchange systems (PC/Mobile).
- Integrated multiple third-party systems with the escrow platform, including Alipay, increasing global revenue by 23% in the first quarter.

Software Engineer, Intern — **NCSoft** — *Seoul, South Korea* — **07/2014 - 08/2014**

- **Fashion Street:** Mobile social game application development (Cocos2D/C++)

Software Engineer, Contract — **Cyberstep** — *Tokyo, Japan* — **06/2013 - 12/2013**

- **Combat Bots Cosmic Commander:** Mobile RTS game client and server development (ActionScript3 & Java)

Education

Bachelor of Science — **Carnegie Mellon University** — *Pittsburgh, PA, USA* — **08/2007 - 12/2010**

- Major in Electrical and Computer Engineering

Projects

- COLORMAN: Creator of a 2D mobile strategy puzzle game (Unity 2D, C#, Android, iOS). Link to YouTube Gameplay **(07/2020)**
- SPIKE: Designed and developed an award-winning action puzzle game SPIKE at a Game Development Competition in Japan **(03/2013)**

Mentorship

- Springboard: Coding Bootcamp Mentor responsible for mentoring and giving career advice to SWE students **(01/2021 - 04/2022)**
- **Computer Science Tutor:** Programming | Data Structure and Algorithms | career advice | coding interview prep | professional portfolio

Others

- **Bronze Award**: Won 4th prize for the development of action game SPIKE at HAL Game Development Competition in Tokyo **(03/2013)**
- **Certificate of Japanese Language Proficiency N1**: The highest-level certificate of Japanese proficiency (24.1% Pass Rate) **(02/2014)**

MOCK INTERVIEW

JOB REFFERAL

Placement