

SUMMARY OF COURSERA COURSE

FRONT-END WEB UI FRAMEWORKS AND TOOLS: BOOTSTRAP 4

RATINGS: /5

WEEK 1 – FRONT-END WEB UI FRAMEWORKS OVERVIEW: BOOTSTRAP

UNIT 1: Course Overview

- Expected background in HTML, CSS and JS
- Web Design consist of : 1) Design (UI, UX, Prototyping, Colors, Graphics and Animation) 2) Deployment, Building and Deployment (MEAN/ MERN stack)
- We will learn about Web Tools like Git, Node, Grunt and Gulp
- Modules Overview

UNIT 2: How to use the Learning Resources

- Course has 4 modules
- Assignment at the end of the module
- Do not just copy and paste code try to understand it
- Learning Resources: 1) Video Lectures 2) Hands Resources 3) Additional Resources
- “1/4th from teacher, 1/4th from own intelligence, 1/4th from classmates, 1/4th only with time (experience)”

UNIT 3: Additional Resources

- Bootstrap site: <https://getbootstrap.com/>

UNIT 4: Full Stack Web Development – The Big Picture – Objectives and Outcomes

- Understand the term “Full Stack” in the context of web development
- Distinguish between front-end, back-end and full stack

UNIT 5: What is Full Stack Web Development

- Front end: client side, HTML, CSS, JS
- Back end: server side, PHP, Java, ASP.NET, Ruby, Python
- The 3 tier architecture of web development: 1) Presentation Layer (concerned with UI related issues) 2) Business Logic Layer (data validation, dynamic content processing) 3) Data Access Layer (data persistence, data access through an API)
- Business Logic Layer: Ruby, Python, Java, C++, PHP are usually used, it interacts with the Data Access Layer (DBMS, Relational Databases), it renders information to the Presentation Layer (server side rendering)
- JSON format is usually explored

UNIT 6: Additional Resources

- Downloaded pdf of lecture “FullStack-BigPicture.pdf”

UNIT 7: Setting up your Development Environment – Objectives and Outcomes

- Set up a Git repo and perform basic Git operations
- Set up and use online Git repos
- Use Node-based modules to perform basic operations

UNIT 8: Setting up your Development Environment

- Don’t delete files after each exercise because the Exercises build on each other, can create a git repo for it

UNIT 9: Setting up Git

- Git is a popular version control system
- Node ecosystem thrives on Git

UNIT 10: Setting up Git (Instructions)

UNIT 11: Basic Git Commands

- “git init”: initializes the current folder as a git repository
- “git checkout <commit_no> <file>”: checkout the file from an older commit

UNIT 12: Basic Git Commands (Instructions)

UNIT 13: Online Git Repositories

- “git remote add origin <URL>”: add the remote online repo
- “git push -u origin master”: push the local fit repo to the origin to the master branch
- Your local repo can only to be matched to one remote repo
- “git clone <repo URL>”: to clone remote repo to local machine

UNIT 14: Online Git Repositories (Instruction)

UNIT 15: Node.js and NPM

- JS was designed as a scripting language for the browser, it has now seen deployment far beyond the browser
- Node.js is based on the JS runtime built on Chrome V8 JS Engine
- Node Architecture
- NPM: manages ecosystem of node modules/ packages

UNIT 16: Setting up Node.js and NPM

UNIT 17: Setting up Node.js and NPM (Instructions)

UNIT 18: Basics of Node.js and NPM

- “package.json”: it serves a documentation for what packages your project depends, it allows you to specify the versions of a packages that your project can use, it makes your build reproducible which means that its way easier to share with other developers
- Initialize package.json: “npm init” and follow the steps
- Install Node Package: “npm install package_name --save-dev”
- “lite-server”, similar to “live server” on VSCode
- “.gitignore”

UNIT 19: Basics of Node.js and NPM (Instructions)

UNIT 20: Additional Resources

- Downloaded “NodeJS.pdf” and “Exercise-Node-NPM.pdf”

UNIT 21: Introduction to Bootstrap – Objectives and Outcomes

- Identify the purpose of using front-end UI frameworks in web design and development
- Set up a project with Bootstrap support
- Configure a web project to use Bootstrap
- Become familiar with the basic features of Bootstrap
- This course covers Bootstrap 4

UNIT 22: Front-end Web UI Frameworks

- Front-end UI Frameworks: are collection of ready to use HTML, CSS and JS templates for UI components (typography, forms, buttons, tables, navigations, dropdowns etc)
- Bootstrap is the most popular Front-end UI Frameworks
- Why FE UI Frameworks: 1) responsive web design 2) cross browser compatibility 3) increased productivity 4) community support

UNIT 23: Introduction to Bootstrap

- First released in 2011

UNIT 24: Getting Started with Bootstrap

- “strap on your boots and lets get started with Bootstrap”
- Downloaded “Bootstrap4-starter.zip”

UNIT 25: Getting Started with Bootstrap (Instructions)

UNIT 26: Additional Resources

UNIT 27: Responsive Design and Bootstrap Grid System – Objectives and Outcomes

- Understand the reasons for using responsive web design in a web project
- Use the Bootstrap Grid system to design responsive websites

- Add your own custom CSS classes to a Bootstrap based Web Project

UNIT 28: Responsive Design

- Mobile First: first satisfy the Mobile constraints
- Bootstrap Grid System, Fluid Images, Media queries
- Media Queries: CSS technology to apply some styles based on the size of the viewport

UNIT 29: Bootstrap Grid System

- Viewport meta tag: ensures that the screen width is set to the device width and the content is rendered with its width in mind
- Bootstrap Grid is designed to be responsive, mobile first, fluid
- Bootstrap Grid system: Container > Row > Col
- Classes to specify screen sizes: default targets all screen sizes from extra small to extra large, “sm”, “md”, “lg”, “xl”
- Class = “row align-items-center”: for vertical alignment
- “justify-content-center”: for horizontal centering

UNIT 30: Responsive Design and Bootstrap Grid System Part 1

UNIT 31: Responsive Design and Bootstrap Grid System Part 1 – Instructions

UNIT 32: Responsive Design and Bootstrap Grid System Part 2

UNIT 33: Responsive Design and Bootstrap Grid System Part 2 – Instructions

UNIT 34: Additional Resources

- Downloaded “Bootstrap-Grid.pdf”

UNIT 35: Assignment 1 Requirements

- Download the unsytlled aboutus.html page

UNIT 36: Assignment 1 –

- Downloaded “aboutus.html”

UNIT 37: Ideation – Objectives and Outcomes

- Clearly express the central idea of your project, identify the problem being addressed
- Delimeate a set of features that you expect your website and app should support
- Identify other projects that might have similar features and would act as exemplars for your project

UNIT 38: Ideation Report Template

- Snap shot
- Might skip honor project

WEEK 2 – BOOTSTRAP CSS COMPONENTS

UNIT 1: Navigation and Navigation Bar – Objectives and Outcomes

- Understand the need for navigation support in a web project
- Use the Bootstrap navigation features including the Navbar and breadcrumbs in providing navigation support in websites
- Use icon fonts for decorating your website with meaningful graphical elements

UNIT 2: Navigation and Navigation Bar

- Most websites are rarely single pages
- Navigation provide visitors an intuitive means of getting around website
- Information Architecture: structure of a system with respect to the way the information is organized, labeled, navigation methods
- Get the pdf of this video
- Don't use generic labels in your Navigation Bar
- Breadcrumbs are like secondary navigation

UNIT 3: Navbar and Breadcrumbs

- Checkout Bootstrap documentation

UNIT 4: Navbar and Breadcrumbs – Instructions

UNIT 5: Icon Fonts

- Icon fonts provide a flexible of including tiny images into out webpages
- They can be used just like regular fonts
- Many icon font packs available
- They are popular lightweight replacement for simple graphics on the website
- Examples of Icon Fonts: 1) Font Awesome 2) Bootstrap Social

UNIT 6: Icon Fonts – Exercise

- Check out font awesome documentation
- You can either use `` or `<i>` for specifying icon fonts

UNIT 7: Icon Fonts – Instructions

UNIT 8: Additional Resources

- Downloaded “Navigation.pdf” and “Icon Fonts.pdf”

UNIT 9: User Inputs – Buttons and Forms – Objectives and Outcomes

- Create and style buttons on a webpage using Bootstrap button classes
- Create and style forms on a web page using Bootstrap form classes

UNIT 10: User Input

- These days on most website users are able to interact with the website to supply information
- User inputs can be facilitated through, hyperlinks, buttons, forms

UNIT 11: Buttons

- Download “contactus.html”

UNIT 12: Buttons – Instructions

UNIT 12: Forms

UNIT 13: Forms – Instructions

UNIT 14: Additional Resources

- Downloaded “User-Input.pdf”

UNIT 15: Displaying Content – Tables and Cards – Objectives and Outcomes

- Present and style tabular data in a table form using Bootstrap support for tables
- Display content using a card on a webpage

UNIT 16: Bootstrap Tables and Cards

- Tables are meant to display tabular data in a web
- Bootstrap cards is a versatile component to display content in myriads of ways

UNIT 17: Displaying Content – Tables and Cards – Exercise

UNIT 18: Displaying Content – Tables and Cards – Instructions

UNIT 19: Additional Content

- Downloaded “Bootstrap-Tables.pdf” and “Bootstrap-Cards.pdf”

UNIT 20: Images and Media – Objectives and Outcomes

- Use images and media and include them in your website
- Support responsive images and media using responsive Bootstrap classes for images and media
- Use thumbnails and media components using Bootstrap classes

UNIT 21: Images and Media

- “a picture is worth a thousand words”
- Bootstrap Image Classes

UNIT 22: Images and Media – Exercise

- Download “img.zip”, unzip into the confusion folder

UNIT 23: Images and Media – Instructions

UNIT 24: Additional Resources

- Downloaded “Images-Media.pdf”

UNIT 25: Alerting Users – Objectives and Outcomes

- Include labels and badges in your web page
- Create, style and include alerts in your web page
- Appreciate the use of progress bars and controlling the state of the progress bars

UNIT 26: Alerting Users

- Badges, Alert messages, Progress Bars
- Badges allow us to show and highlight little information on our site

UNIT 27: Alerting Users – Exercise

UNIT 28: Alerting Users – Instructions

UNIT 29: Additional Resources

- Downloaded “Alerting-Users.pdf”
- Checkout bootstrap documentation on Badges, Alerts and Progress

UNIT 30: Assignment 2

- This assignment will explore understanding of all the topics covered
- 3 tasks to complete

UNIT 31: Additional Resource

- You have to marry Bootstrap Documentation page

UNIT 32: UI Design and Prototyping – Objectives and Outcomes

- Construct a wireframe diagram to visually represent the structure of your UI
- Construct a prototype to enable understanding the flow of your application

UNIT 33: UI Design and Prototyping Report Template

- Snap shots

UNIT 34: Additional Resources

- Links to resources for UI design
- Not ready for UI/UX design yet

WEEK 3 – BOOTSTRAP JAVASCRIPT COMPONENTS

UNIT 1: Bootstrap JavaScript Components – Objectives and Outcomes

- Understand the various Bootstrap components that require JS support in order to function
- Use the data-* attributes that Bootstrap's JS API provides for you to control the JS components without writing a single line of JS code

UNIT 2: Bootstrap JS Components

- Base class > births modifier classes
- JS components in Bootstrap can be used without writing a single line of JS

UNIT 3: Additional Resources

- Downloaded “Bootstrap-JS.pdf”

UNIT 4: Tabs and Tabbed Navigation – Objectives and Outcomes

- Design a tabbed navigation for your content
- Use tab panes to organize the content and navigate the content using tabbed navigation

UNIT 5: Tabs, Pills and Tabbed Navigation

- Tabs and Pills used for navigating content within a page that is organized into multiple panes

UNIT 6: Tabs – Exercise

UNIT 7: Tabs – Instructions

UNIT 8: Additional Resources

- Downloaded “Tabs-Pills-Navigation.pdf”

UNIT 9: Hide and Seek – Objectives and Outcomes

- Use the collapse plugin to hide and reveal content
- Construct the accordion using cards

UNIT 10: Collapse and Accordion

- Collapse plugin provides a quick way of revealing and hiding content
- Accordion named derived from the behavior of the musical instrument “accordion”

UNIT 11: Accordion – Exercise

UNIT 12: Accordion – Instructions

UNIT 13: Additional Resources

- Downloaded “Collapse.pdf”

UNIT 14: Revealing Content – Objectives and Outcomes

- Set up a tooltip to be displayed when the user hovers over an area of the page
- Enable popovers when the user clicks on a link or button
- Reveal and hide modals when the user clicks on a link or button

UNIT 15: Tooltips, Popovers and Modals

- Revealing content upon interacting with an element on the web page
- Info is displayed as an overlay
- You have to click on the element to activate a pop-over
- Model can present more detailed information, example is the login form pop up

UNIT 16: Tooltips and Modals – Exercise

- Bootstrap advises that all “modal” related code should be placed at the top

UNIT 17: Tooltips and Modals – Instructions

UNIT 18: Additional Resources

- Downloaded “Tooltips-Popovers-Modals.pdf”

UNIT 19: Carousel – Objectives and Outcomes

- Use a carousel component in your web page
- Configure various aspects of the carousel
- Add controls to the carousel to manually control it

UNIT 20: Carousel

- Allows you to display multiple items on your web page, like a slide show

UNIT 21: Carousel – Exercise

UNIT 22: Carousel – Instructions

UNIT 23: Additional Resources

- Downloading “Carousel.pdf”

UNIT 24: Assignment 3

- Make the orders form a modal

WEEK 4 – WEB TOOLS

UNIT 1: Bootstrap and JQuery – Objectives and Outcomes

- Understand how to use jQuery and JS and Bootstrap JS component methods to control the behavior of the components
- Write JS code taking advantage of the Bootstrap JS component methods and jQuery methods for controlling Bootstrap JS components

UNIT 2: Bootstrap and jQuery

- Bootstrap JS plugins are mostly written based on jQuery
- JS components can be used without writing a single line of JS
- JQuery is a light weight JS library, DOM manipulation, CSS manipulation, event methods, effects and animations, AJAX
- “\$(selector).action()” > jQuery syntax
- JQuery selectors: HTML element, using the #id, using the class (“.btn”), using the attribute, current element (\$(this))
- JQuery events: Mouse (click, dblclick), Keyboard (keypress), Form (submit, change, focus, blur), Document, Window
- JQuery event methods: ready(), click(), on()

UNIT 3: Bootstrap and jQuery – Exercise

UNIT 4: Bootstrap and jQuery –Instructions

UNIT 5: More Bootstrap and jQuery – Exercise

UNIT 6: More Bootstrap and jQuery – Instructions

UNIT 7: Additional Resources

- Downloaded “Bootstrap-jQuery.pdf”

UNIT 8: CSS Preprocessors – Objectives and Outcomes

- Write Less and Sass code to define your CSS classes
- Compile the Less and Sass code into the corresponding CSS classes

UNIT 9: CSS Preprocessors – Less and Sass

- Bootstrap is built using SASS for its source
- CSS is limited when it comes to features like defining variables, nesting selectors, expressions, functions
- Bootstrap 3 used LESS, Bootstrap 4 migrated to use SASS
- Preprocessors provides programming-like syntax
- Preprocessors Features: variables, mixins, function expressions, nesting
- Mixins can hold multiple CSS declarations unlike variables

UNIT 10: LESS – Exercise

- “npm install –g [less@2.7.2](#)”

UNIT 11: LESS –Instructions

UNIT 12: SCSS – Exercise

- “npm install –save-dev [node-sass@4.7.2](#)”
- You are advised not to taper with Bootstrap source codes until you are a pro

UNIT 13: SCSS - Instructions

UNIT 14: Additional Resources

- Downloaded “CSS-Preprocessors.pdf”

UNIT 15: Assignment 4

UNIT 16: Building and Deployment – Objectives and Outcomes

- Configure NPM scripts and automate your web development
- Prepare your project for being hosted on a web server

UNIT 17: Building and Deployment

- Web development and deployment involves a lot of repetitive tasks
- DRY principle: do not repeat yourself
- Need for build tools to automate tasks
- JSHint: checking JS code for errors and potential problems (static code analysis)
- Uglification: minification + mangling (reduce local variables to single letters)

UNIT 18: NPM Scripts

- VSCode extensions have replaced the use of “npm script” substantially
- NPM Scripts are supported through the “scripts” property in the package.json file

UNIT 19: NPM Scripts Part 1 – Exercise

- npm install --save-dev [onchange@3.3.0](#) [parallelshell@3.0.2](#)
- “onchange”: watches for any changes in our project files and trigger scripts for any change
- “parallelshell”: allows us to run multiple scripts at once

UNIT 20: NPM Scripts Part 1 – Instructions

UNIT 21: NPM Scripts Part 2 – Exercise

- Combine all the JS/ CSS files into one, it reduces the load time
- “npm install --save-dev [rimraf@2.6.2](#)”
- “npm –g install [copyfiles@2.0.0](#)”

- “npm install –g [imagemin-cli@3.0.0](#)”: to compress images
- Might use VSCode extensions for this process
- “npm install –save-dev [usemin-cli@0.5.1](#) [cssmin@0.4.3](#) [uglifyjs@2.4.11](#) [htmlmin@0.0.7](#)”: to minimize html, css and js files

UNIT 22: NPM Scripts Part 2 – Instructions

UNIT 23: Additional Resources

- Downloaded “Building-Deployment.pdf” and “NPM-Scripts.pdf”

UNIT 24: Building and Deployment – Task Runners – Objectives and Outcomes

- Configure Grunt tasks and automate your web development using Grunt
- Define Gulp task in code to automate the web development using Gulp

UNIT 25: Task Runner

- Npm scripts can be used to automate some task in web development
- Task Runners: configure the task once, and rerun it automatically as many times as needed
- Several task runners: Grunt, Gulp, Cake, Brunch, Broccoli (obsession with food)
- Grunt: configuration over code
- Gulp: code over configuration

UNIT 26: Grunt Part 1 – Exercise

- “Time to start grunting”
- “npm install [–grunt-cli@1.2.0](#)”: installing grunt CL globally
- “npm install [grunt@1.0.2](#) –save-dev”
- Create a new file “Gruntfile.js”
- “npm install –save-dev grunt-sass@2.1.0 ”
- “npm install –save-dev [time-grunt@1.4.0](#) [jit-grunt@0.10.0](#)”
- “npm install –save-dev [grunt-contrib-watch@1.0.0](#)”
- “npm install –save-dev [grunt-browser-sync@2.2.0](#)”
- Read the documentation for different plugins to know how to configure them
- “grunt” on the terminal to run “default” task
- The “watch” task should always be last in the sequence of tasks

UNIT 27: Grunt Part 1 – Instructions

UNIT 28: Grunt Part 2 – Exercise

- “npm install –save-dev grunt-contrib-copy@1.0.0 ”
- “npm install –save-dev grunt-contrib-clean@1.1.0”
- Read documentations for Grunt Plugins
- “npm install grunt-contrib-imagemin@2.0.1 --save-dev”

- npm install grunt-contrib-concat@1.0.1 --save-dev
- npm install grunt-contrib-cssmin@2.2.1 --save-dev
- npm install grunt-contrib-htmlmin@2.4.0 --save-dev
- npm install grunt-contrib-uglify@3.3.0 --save-dev
- npm install grunt-filerev@2.3.1 --save-dev: adds additional extension to new files (file revision)
- npm install grunt-usemin@3.1.1 --save-dev
- Stackoverflow is a needed resource
- Grunt “htmlmin” has to be performed after “usemin”

UNIT 29: Grunt Part 2 – Instructions

UNIT 30: Gulp Part 1 – Exercise

- “npm install -g [gulp-cli@2.0.1](#)”
- “npm install gulp@3.9.1 --save-dev”
- “npm install gulp-sass@3.1.0 browser-sync@2.23.6 --save-dev”
- Create “gulpfile.js”
- Gulp operates on streams; you stream files through the pipe
- Gulp Streams: “gulp.src()” > “gulp.pipe()” > “gulp.dest()”

UNIT 31: Gulp Part 1 – Instructions

UNIT 32: Gulp Part 2 – Exercise

- “npm install del@3.0.0 --save-dev”
- “npm install gulp-imagemin@4.1.0 --save-dev”
- “npm install gulp-uglify@3.0.0 gulp-usemin@0.3.29 gulp-rev@8.1.1 gulp-clean-css@3.9.3 gulp-flatmap@1.0.2 gulp-htmlmin@4.0.0 --save-dev”
- “flatmap”: takes multiple html files and works on them in parallel (same set of pipes)
- “rev”: similar functions to filerev on Grunt

UNIT 33: Gulp Part 2 – Instructions

UNIT 34: Additional Resources

- Downloaded “Task-Runners.pdf”
- <https://www.dbswebsite.com/blog/the-advantages-of-using-task-runners/>
- <https://gulpjs.com/>
- <https://gruntjs.com/>

UNIT 35: Conclusions

- Together with UI frameworks, JS frameworks facilitate Responsive Front-end Development, “Conclusion.pdf”