|  |  |
| --- | --- |
| Concept | Description |
| 1. Domain Name resolution (DNS) | Editing the Hosts file to redirect our local request of [www.slacker.com](http://www.slacker.com) to 35.225.25.255 |
| 1. TCP/IP Protocols | Handles the data transmission over the internet which allows the packets from the server, arrive at the correct host computer, complete, and in the right order to be able to view the web page. |
| 1. Front-end Developer role | Building the website using HTML, CSS, and JavaScript allowed us to get a feel of what a Front-end Developer would go through. The use of different tools i.e. Sublime, Web browser, Git, etc. |
| 1. Presentation tier | Built 3 webpages that link together via its user interface (login and navigation) where the user understands what the main function of where they are. |
| 1. HTML | Code in HTML Semantic, and understand that the HTML portion is just the content (not stylized). This allows modularity in developing to separate the styling and interactivity. Although you can include all 3 in the same HTML file, best practice is to separate. |
| 1. CSS | Code in CSS, to style the content we are displaying to the user i.e. adding the different colours, padding, centering, etc. |
| 1. JavaScript | Code in JavaScript, to be able read a JSON object and fill in content dynamically i.e. loading user messages. |
| 1. CSS selectors | How to style specific elements either by selecting the element, creating a class, or and ID. |
| 1. Agile Methodology | Learning the Agile methodology and applying it to project by creating essentially 3 sprints with each sprint delivering a usable product and in the end connecting them to work as one. |
| 1. User Stories (Scrum) | Assigned the mini projects through user stories and using the terminology like Scrum Master and Product Owner. |
| 1. Unit Testing | As we build the different parts of the webpage, we need to unit test code before we integrate |
| 1. Troubleshooting | This goes hand in hand with unit testing as we enviably find or do something wrong where we need to identify the root problem, and fix one thing at a time to not get confused or lose what fix applied fix the problem. |
| 1. Responsive Web Design | The use of fluid grids and media queries to be flexible to the screen the end user is using. |
| 1. Principle of Focused Navigation | Have a contextual navigation that is clean and simple |
| 1. Text Editors & IDEs | As mentioned previously we used a popular text editor among web developers Sublime and gave us a feel for what is being used in the field today. |
| 1. Version Control (Git) | Used the widely used standard for version control Git and Github to allows to make changes safely and create a history of version did. |
| 1. Principals of Virtualization | Used Oracle VM to create a virtual instance of Ubuntu |
| 1. Deployment to Unix-like Systems | Connected to a Bridge, then to a server, and deployed our web application using CLI |
| 1. DevOps Role | Played the part of being able to Build, Test, Release our code to the cloud. |
| 1. Hypervisors | Installed and used Hypervisor Type 2 in order to use Ubuntu on top of my Windows machine. |