Full Stack Web Developer Syllabus

Contact Info

While going through the program, if you have questions about anything, you can reach us at enterprise-support@udacity.com. For help from Udacity Mentors and your peers visit the Udacity Classroom.

Nanodegree Program Info

Students will learn about building out the infrastructure that powers and supports the many web, desktop, mobile and integrated applications in the world.

Prerequisite Skills

A well-prepared learner is able to:

- Recommended having experience programming in HTML, CSS, and Python
- *If you don't have any experience programming our Intro to Programming or Front End Web Developer programs are great introductory courses.

Required Hardware

- Webcam
- Microphone
- 64-bit computer

Required Software

- Sublime Text 2.0.2+
- Atom 0.177.0
- MacOS or OSX
- Gitbash
- Git 2.2.1
- Python 2.7/3.6
- VirtualBox 4.2.28
- Vagrant 1.7.2

Version: 1.0.0

Length of Program: 103 Days*

** This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.*

Part 1: Welcome to the Program!

Welcome to the Full Stack Web Developer Nanodegree program. This is your first step on your journey to become a Full Stack Developer. Learn what this program is all about as well as how to find support along your learning journey.

Part 2: Developer's Tools

Brush up your knowledge of essential developers' tools such as the Unix shell, Git, and Github; then apply your skills to investigate HTTP, the Web's fundamental protocol.

Part 3: Databases with SQL and Python

Master SQL databases and build multi-user web applications using the Flask framework, SQLAlchemy, and authentication providers such as Google and Facebook.

Project: Project: Logs Analysis

In this project, you'll practice your SQL skills by building a reporting tool that summarizes data from a large database.

Supporting Lessons

Lesson	Summary
Data and Tables	Learn the principles behind relational data organization: tables, queries, aggregations, keys, and joins.
Elements of SQL	Start learning SQL by using the select and insert statements to read and write data in database tables.
Python DB-API	Learn the Python database API, and apply your knowledge to fix common bugs that arise in database-backed web services.
Deeper Into SQL	Create your own database tables using normalized table design, using keys to declare relationships between tables; then apply these relationships to draw conclusions from data.

Part 4: Servers, Authorization and CRUD

Learn the CRUD pattern (Create, Read, Update, Delete) and how it relates to RESTful architectures and to the operations of a database-backed web service. Learn the difference between authentication and authorization and some best practices in developing a login system.

Project: Project: Item Catalog

You will develop an application that provides a list of items within a variety of categories as well as provide a user registration and authentication system.

Supporting Lessons

Lesson	Summary
Working with CRUD	Learn the CRUD pattern (Create, Read, Update, Delete) and how it relates to RESTful architectures and to the operations of a database-backed web service.
Making a Web Server	Build a web service in Python that uses your database to implement CRUD operations.
Developing with frameworks	Apply the Flask framework in Python to build web services more quickly and reliably.
Iterative Development	Learn the basics of agile and iterative development while building a restaurant menu application.
Authentication vs Authorization	Learn the difference between authentication and authorization and some best practices in developing a login system.
Creating Google Sign in	Investigate OAuth and build third-party sign-in into your web applications using Google's authentication services.
Local Permission System	Use a local permission system to protect pages based on each user, not just whether that user is logged in.
Adding Facebook and other providers	Discover additional OAuth providers and add Facebook authentication in your application, giving more choices for third-party auth.
What's and Why's of APIs	Discover the foundation of Web APIs, the common systems for passing and updating information to web applications.
Accessing Published APIs	Practice gathering and reading information from web APIs that already live on the internet.
Creating your own API Endpoints	Begin creating your own API endpoints and learn to serialize your data to package it up for transfer across the web using HTTP.
Securing your API	Control access to your APIs by limiting who can access the resources behind them and ensuring that only authorized users can read and modify data.
Writing Developer-Friendly APIs	Customize your APIs to be usable and approachable by the developers who will consume them.

Part 5: Deploying to Linux Servers

You will take a baseline installation of a Linux distribution on a virtual machine and prepare it to host your web applications, to include installing updates, securing it from a number of attack vectors, and installing and configuring web and database servers.

Project: Project: Linux Server Configuration

You will take a baseline installation of a Linux distribution on a virtual machine and prepare it to host your web applications.

Supporting Lessons

Lesson	Summary
Intro to Linux	Gain an understanding of the Linux operating system and how it differs from other operating systems you may have experienced in the past.
Linux Security	Dive deep into Linux Security to ensure your service remains stable and free from attackers.
Web Application Servers	Install all of the required software to turn your Linux server into a full-fledged web application server and host your very own application!

Part 6: Congratulations! What's Next?

Great work finishing the program! Here's some advice on next steps in your programming journey.

Udacity

Generated Sat Jan 9 02:36:46 PST 2021