UDACITY
FOR ENTERPRISE

THE SCHOOL OF PROGRAMMING AND DEVELOPMENT

Full Stack JavaScript Developer

NANODEGREE SYLLABUS

Overview

Full Stack JavaScript Developer Nanodegree Program

The goal of the Full Stack JavaScript Developer Nanodegree program is to equip learners with everything a developer needs to build rich experiences delivered with the web using a modern architecture and technology stack. Learners will cover everything from building UI and UX to creating APIs and server side business logic, and developing the persistence layer to store, process and retrieve data.

A graduate of this program will be able to:

- Build client-side experiences and applications using Angular, collecting data from users and from backends, providing rich user interactions and organizing code and data.
- Build server-side executed code with TypeScript and integrate with 3rd party code such as Angular's Server Side Rendering.
- Leverage Express.js to architect and build APIs that power dynamic functionality and to generate and supply data to web and mobile clients.
- Persist data to a database, query and retrieve data and pass this data all the way through to various client devices.

Program Information

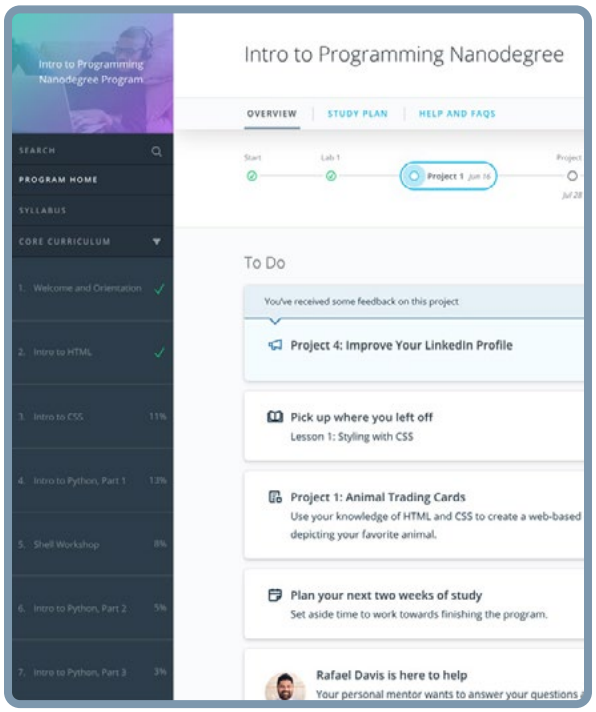
TIME
4 months
Study 10 hours/week

LEVEL
Practitioner

PREREQUISITES
- Fundamental programming concepts
- JavaScript object-oriented programming features
- HTML and CSS
- DOM events
- Simple programs in JavaScript
- JSON object

LEARN MORE ABOUT THIS NANODEGREE
Contact us at enterpriseNDs@udacity.com.

Our Classroom Experience



REAL-WORLD PROJECTS

Learners build new skills through industry-relevant projects and receive personalized feedback from our network of 900+ project reviewers. Our simple user interface makes it easy to submit projects as often as needed and receive unlimited feedback.

KNOWLEDGE

Answers to most questions can be found with Knowledge, our proprietary wiki. Learners can search questions asked by others and discover in real-time how to solve challenges.

LEARNER HUB

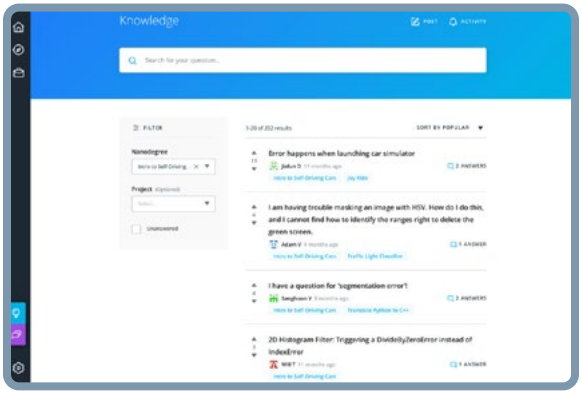
Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

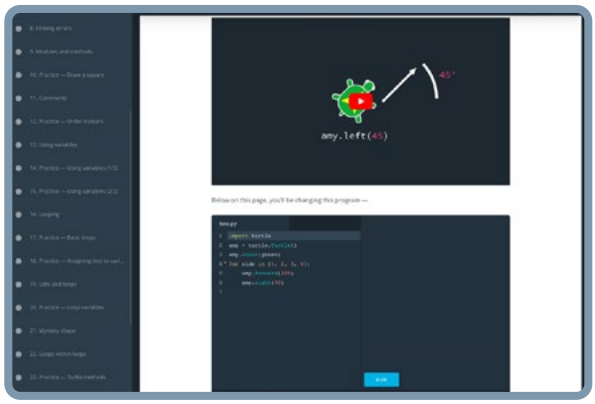
WORKSPACES

Learners can check the output and quality of their code by testing it on interactive workspaces that are integrated into the classroom.

QUIZZES

Understanding concepts learned during lessons is made simple with auto-graded quizzes. Learners can easily go back and brush up on concepts at anytime during the course.





CUSTOM STUDY PLANS

Mentors create a custom study plan tailored to learners' needs. This plan keeps track of progress toward learner goals.

PROGRESS TRACKER

Personalized milestone reminders help learners stay on track and focused as they work to complete their Nanodegree program.

Learn with the Best

Guillaume Bibeau Laviolette

SOFTWARE ENGINEER

Guillaume is a software developer that has worked for companies such as Shutterstock and Filevine as a software and cloud engineer. He obtained his bachelor's of statistics and probability at McGill University in Canada.

Rachel Manning

FULL STACK DEVELOPER

Rachel is a full stack freelance developer and educator. As an advocate for continued learning, she is passionate about mentoring women and the underserved community in technology.

Alyssa Hope

SOFTWARE ENGINEER

Alyssa is a full stack developer who was previously the lead instructor at a coding bootcamp. With a degree in International Communications, her passion is to express thoughts well, whether in code or writing.

Andrew Wong

FULL STACK ENGINEER

Andrew is a full stack engineer who enjoys making the world a better place through code. He first discovered his passion for teaching as an instructor at App Academy, and continues to enjoy empowering students to advance their education.

Nanodegree Program Overview

Course 1: Backend Development with Node.js

There are quite a few technologies involved to build the backend of an application that's enterprise ready. This course introduces the fundamental tools needed to build a basic API in a way that is both scalable, and maintainable. The course will go through working with Node.js and the core modules available, writing TypeScript for developer error reduction, testing with Jasmine to introduce unit testing in a Test Driven Development environment and working with Express as a framework for building APIs.

Project: Image Processing API

Students will be designing an API for image processing that allows the user to visit a url and using url parameters, resize the image based on the parameters provided. Upon viewing an image that's already been resized, a cached image will be served. This is the core functionality seen in placeholder image websites and can be implemented with a frontend to better serve appropriately sized images. The API presents the first opportunity to pull together the technologies of the course and tie them together in a commonly used application.

LESSON TITLE	LEARNING OUTCOMES
GETTING STARTED WITH NODE.JS	• Understand why to use Node.js. • Install and understand how Node.js is updated. • Understand the event loop and control how asynchronous code is processed. • Use Node.js REPL to write js expressions, then just Node.js to run a js file. • Extend JavaScript by using global variables and functions not available in the browser. • Use NPM init to create a package.json/lock to install dependencies and configure scripts.

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
DEVELOPING WITH TYPESCRIPT	• Take advantage of strict typing to reduce error by installing TypeScript to work with Node.js. • Create valid, formatted, readable TypeScript by configuring ESLint/ Prettier to work with TypeScript. • Type variables, functions and objects with TypeScript. • Manage async/await, promises and error handling with TypeScript
UNIT TESTING WITH JASMINE	• Install Jasmine and configure it to work with JavaScript after TypeScript has been compiled. • Organize, write and run unit tests. • Create asynchronous tests and use Supertest to perform endpoint tests.
BUILDING A SERVER	• Build a server by applying the top features of Express's root app object. • Improve an application by creating and applying middleware. • Take advantage of the file system by learning to write and read files from disk.

Nanodegree Program Overview

Course 2: Creating an API with PostgreSQL and Express

This course covers the primary skills required for API development. Students will build a RESTful JSON API with Node and Postgres. Along the way, you will cover essential topics like databases and querying, API architecture, database migrations, REST, CRUD, creating a testing environment, password hashing and route authorization via JWTs. By completing the exercises and course content, students will gain the knowledge to create a secure, well organized API from scratch and learn skills JavaScript developers use every day.

Project: Build a Storefront Backend

Students will imagine themselves as a full stack developer at a small company, asked to craft an API to meet a set of requirements created by business stakeholders and to be consumed by a front end developer coworker. You will use skills and understanding from the course to discern the best architecture, endpoint structure and database schema to complete the task. Modeled after a workplace environment and true to life task, in this project you will demonstrate that you can reason through the design of an API and are capable of writing the code and logic necessary to meet requirements.

LESSON TITLE	LEARNING OUTCOMES
DATABASES AND SQL	• Understand what databases are. • Gain visibility into the most popular types of databases, their respective strengths and use cases. • Become familiar with Relational Databases. • Connect to a database with interactive terminal psql. • Write basic database queries with SQL. • Create, structure and fill a Postgres database. • Know what CRUD is and how it pertains to database actions. • Understand when and how to use SQL filters and foreign keys.

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
CREATING AN API WITH POSTGRES CONNECTION	• Set up a Node application to connect to a postgres database. • Install a Node package dotenv for environment variables. • Learn best practices for using environment variables. • Create Models in Node. • Understand the need for database migrations. • Install a Node package for database migrations. • Set up a testing environment with a separate testing database. • Testing models with Jasmine.
CREATING AN API WITH EXPRESS	• Understand RESTful APIs and how to design one. • Understand the full stack request/response loop through the API. • Be introduced to what CORS is and when it is necessary. • Write Express custom middleware. • Write clean, well organized Express routes that mirror API structure.
AUTHENTICATION & AUTHORIZATION IN A NODE API	• Be familiar with password hashing, salt and pepper. • Understand why these practices are important for data security. • Install a Node package Bcrypt and learn to set up Bcrypt for password hashing at user creation. • Understand what JWTs are and how they work. • Implement route authorization with JWTs.
SQL FOR ADVANCED API FUNCTIONALITY	• Understand database relationships and when to implement them. • Create RESTful routes that demonstrate relationships between data via well organized routes. • Write custom model and handler methods for extending API endpoints. • Understand SQL Joins and result ordering. • Write informational API endpoints to support dashboard functionality. • Organize more advanced business logic in a Node API for cleaner code and separation of concerns.

Nanodegree Program Overview

Course 3: Angular Fundamentals

In Angular Fundamentals, students will learn the most important and foundational skills for building Single Page Applications (SPAs). You will discover the architecture of an application, explore how to retrieve and flow data throughout an application, and see how applications scale in a maintainable and performant way. Upon completion of the course, you will be able to build new and expand existing Angular applications with new components and features, architect an Angular application for clarity and maintainability while following best practices and create and use dependencies such as services and third-party libraries to enrich and extend applications.

Project: My Store

In this project, students will build a full single-page ecommerce application with Angular called MyStore. Your application will contain a variety of different Angular components that communicate with each other, such as a product list component that renders a list of items for which a user can shop. You'll pull this data by making requests to a backend API, then populate your page with items that can be added to the shopping cart.

You'll build and nest these components in a logical structure for optimal navigation and routing, such as bringing users to a product detail page. Through using services, among other tools, you'll be able to share data with any component that needs it, such as your shopping cart. Your application will also be able to handle and respond to user input, through Angular's powerful template-driven forms.

LESSON TITLE	LEARNING OUTCOMES
ANGULAR OVERVIEW	• Set up and install the Angular CLI. • Scaffold a new application using the Angular CLI. • Generate a component. • Organize components into closely-related sets of capabilities (i.e., modules).

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
COMPONENTS	• Build a component by defining a class containing application data and logic. • Build templates to represent an application's user interface. • Use data binding to re-render the application by detecting changes in state. • Extend the style or behavior of HTML elements with directives. • Use the Angular router to handle navigation between views.
LIBRARIES & SERVICES	• Extend Angular's base functionality by importing libraries and modules. • Use dependency injection to provide custom services and capabilities. • Find, add and manage an application's dependencies.
DATA	• Fetch data via HTTP with the HTTP Client Module. • Collect user input by building template-driven forms. • Validate user input.

Nanodegree Program Overview

Course 4: Deployment Process

Being able to deploy your own application is a skill that is often overlooked by developers, thus making it a rare and valuable skill to have! This course will teach the necessary knowledge to create your own production environment and automate the deployment of code to it. By building an automated pipeline and scripts students will gain insights into the world of automated deployments that has been revolutionizing how fast companies are able to deliver features to their customers.

Project: Hosting a Full Stack Application

Students will be hosting and creating an automated deployment pipeline for a full stack application. You will have the option of using your own project or using a provided full stack application. During the project students will provision AWS infrastructure and prepare the application for hosting it. After this is done, you will move on to creating different scripts for deploying the application and creating a deployment pipeline. This project will expose students to a key process that is present on most software development projects.

LESSON TITLE	LEARNING OUTCOMES
SETTING UP A PRODUCTION ENVIRONMENT	• Create and configure a Postgres database in RDS. • Start an elastic beanstalk environment using the pre-configured node template. • Create environment properties in Elastic Beanstalk . • Create an S3 bucket and configure it for web hosting.
INTERACT WITH CLOUD SERVICES	• Ensure an application is healthy by using the Elastic Beanstalk CLI. • Apply code changes in an S3 bucket by using the AWS CLI to update the bucket content. • Apply code changes in Elastic Beanstalk by using the EB CLI to deploy a new application version.

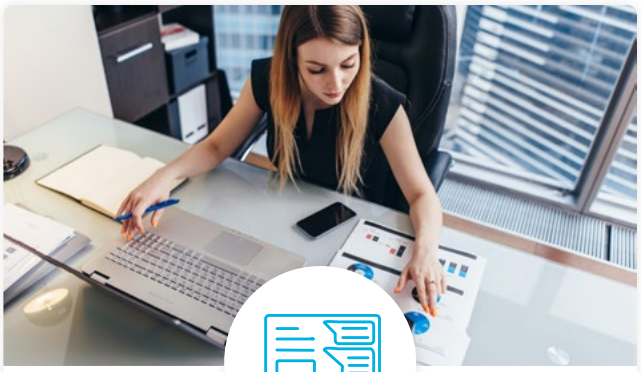
Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
WRITE SCRIPTS FOR WEB APPLICATIONS	• Create scripts to build back-end and front-end applications. • Create scripts to deploy different pieces of a full stack application. • Create scripts that ensure a build follows quality standards. • Create a root level script to install all application dependencies.
CONFIGURE AND DOCUMENT A PIPELINE	• Connect a repository to CirceCi. • Create a manual approval step for deployments. • Create an automated integration process. • Create documentation using markdown files and diagrams. • Create an automated deployment process.

Our Nanodegree Programs Include:

Pre-Assessments

Our in-depth workforce assessments identify your team's current level of knowledge in key areas. Results are used to generate custom learning paths designed to equip your workforce with the most applicable skill sets.



Dashboard & Progress Reports

Our interactive dashboard (enterprise management console) allows administrators to manage employee onboarding, track course progress, perform bulk enrollments and more.

Industry Validation & Reviews

Learners' progress and subject knowledge is tested and validated by industry experts and leaders from our advisory board. These in-depth reviews ensure your teams have achieved competency.

Real World Hands-on Projects

Through a series of rigorous, real-world projects, your employees learn and apply new techniques, analyze results, and produce actionable insights. Project portfolios demonstrate learners' growing proficiency and subject mastery.

Our Review Process

Real-life Reviewers for Real-life Projects

Real-world projects are at the core of our Nanodegree programs because hands-on learning is the best way to master a new skill. Receiving relevant feedback from an industry expert is a critical part of that learning process, and infinitely more useful than that from peers or automated grading systems. Udacity has a network of over 900 experienced project reviewers who provide personalized and timely feedback to help all learners succeed.

Vaibhav
UDACITY LEARNER

"I never felt overwhelmed while pursuing the Nanodegree program due to the valuable support of the reviewers, and now I am more confident in converting my ideas to reality."

now at

CODING VISIONS INFOTECH

All Learners Benefit From:

- Line-by-line feedback for coding projects
- Industry tips and best practices
- Advice on additional resources to research
- Unlimited submissions and feedback loops

How it Works

Real-world projects are integrated within the classroom experience, making for a seamless review process flow.

- Go through the lessons and work on the projects that follow
- Get help from your technical mentor, if needed
- Submit your project work
- Receive personalized feedback from the reviewer
- If the submission is not satisfactory, resubmit your project
- Continue submitting and receiving feedback from the reviewer until you successfully complete your project

About our Project Reviewers

Our expert project reviewers are evaluated against the highest standards and graded based on learners' progress. **Here's how they measure up to ensure your success.**

900+

Expert Project Reviewers

Are hand-picked to provide detailed feedback on your project submissions.

1.8M

Projects Reviewed

Our reviewers have extensive experience in guiding learners through their course projects.

3

Hours Average Turnaround

You can resubmit your project on the same day for additional feedback.

4.85 /5

Average Reviewer Rating

Our learners love the quality of the feedback they receive from our experienced reviewers.

UDACITY
FOR ENTERPRISE

2440 W El Camino Real, #101
Mountain View, CA 94040, USA - HQ

For more information visit: www.udacity.com/enterprise