



HOLY ANGEL UNIVERSITY
SCHOOL OF COMPUTING
1st Semester, School Year 2025-2026



6WC SERVER freeCodeCamp

Finals Badging

Submitted by:

Charles Daniel B. Garcia

Submitted to:

Mr. Tjakoen Stolk

WD 302

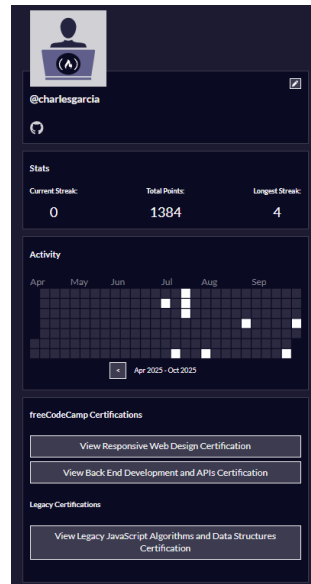
October 2, 2025



HOLY ANGEL UNIVERSITY
SCHOOL OF COMPUTING
1st Semester, School Year 2025-2026



freeCodeCamp Profile Link: <https://www.freecodecamp.org/charlesgarcia>



Legacy JavaScript Algorithms and Data Structures

Basic Algorithm Scripting

Basic Algorithm Scripting

An algorithm is a series of step-by-step instructions that describe how to do something.

To write an effective algorithm, it helps to break a problem down into smaller parts and think carefully about how to solve each part with code.

In this course, you'll learn the fundamentals of algorithmic thinking by writing algorithms that do everything from converting temperatures to handling complex 2D arrays.

▼ Collapse course

16/16

- ✔ Convert Celsius to Fahrenheit
- ✔ Reverse a String
- ✔ Factorialize a Number
- ✔ Find the Longest Word in a String
- ✔ Return Largest Numbers in Arrays
- ✔ Confirm the Ending
- ✔ Repeat a String Repeat a String
- ✔ Truncate a String
- ✔ Finders Keepers
- ✔ Boo who
- ✔ Title Case a Sentence
- ✔ Slice and Splice
- ✔ Falsy Bouncer
- ✔ Where do I Belong
- ✔ Mutations
- ✔ Chunky Monkey



HOLY ANGEL UNIVERSITY SCHOOL OF COMPUTING

1st Semester, School Year 2025-2026



Functional Programming

Functional Programming

Functional Programming is another popular approach to software development. In Functional Programming, code is organized into smaller, basic functions that can be combined to build complex programs.

In this course, you'll learn the core concepts of Functional Programming including pure functions, how to avoid mutations, and how to write cleaner code with methods like `_map()` and `_filter()`.

▼ Collapse course 24/24

- Learn About Functional Programming
- Understand Functional Programming Terminology
- Understand the Hazards of Using Imperative Code
- Avoid Mutations and Side Effects Using Functional Programming
- Pass Arguments to Avoid External Dependence in a Function
- Refactor Global Variables Out of Functions
- Use the `map` Method to Extract Data from an Array
- Implement `map` on a Prototype
- Use the `filter` Method to Extract Data from an Array
- Implement the `filter` Method on a Prototype
- Return Part of an Array Using the `slice` Method
- Remove Elements from an Array Using `slice` Instead of `splice`
- Combine Two Arrays Using the `concat` Method
- Add Elements to the End of an Array Using `concat` Instead of `push`
- Use the `reduce` Method to Analyze Data
- Use Higher-Order Functions `map`, `filter`, or `reduce` to Solve a Complex Problem
- Sort an Array Alphabetically using the `sort` Method
- Return a Sorted Array Without Changing the Original Array
- Split a String into an Array Using the `split` Method
- Combine an Array into a String Using the `join` Method
- Apply Functional Programming to Convert Strings to URL Slugs
- Use the `every` Method to Check that Every Element in an Array Meets a Criteria
- Use the `some` Method to Check that Any Elements in an Array Meet a Criteria
- Introduction to Currying and Partial Application

Intermediate Algorithm Scripting

Intermediate Algorithm Scripting

Now that you know the basics of algorithmic thinking, along with OOP and Functional Programming, test your skills with the Intermediate Algorithm Scripting challenges.

▼ Collapse course 21/21

- Sum All Numbers in a Range
- Diff Two Arrays
- Seek and Destroy
- Wherefore art thou
- Spinal Tap Case
- Pig Latin
- Search and Replace
- DNA Pairing
- Missing letters
- Sorted Union
- Convert HTML Entities
- Sum All Odd Fibonacci Numbers
- Sum All Primes
- Smallest Common Multiple
- Drop it
- Steamroller
- Binary Agents
- Everything Be True
- Arguments Optional
- Make a Person
- Map the Debris



JavaScript Algorithms and Data Structures Projects

JavaScript Algorithms and Data Structures Projects

This is it — time to put your new JavaScript skills to work. These projects are similar to the algorithm scripting challenges you've done before – just much more difficult.

Complete these 5 JavaScript projects to earn the JavaScript Algorithms and Data Structures certification.

Palindrome Checker	✓
Roman Numeral Converter	✓
Caesars Cipher	✓
Telephone Number Validator	✓
Cash Register	✓

(view the timeline in my profile to see the source code)

Back End Development and APIs

MongoDB and Mongoose

MongoDB and Mongoose

MongoDB is a database application that stores JSON documents (or records) that you can use in your application. Unlike SQL, another type of database, MongoDB is a non-relational or "NoSQL" database. This means MongoDB stores all associated data within one record, instead of storing it across many preset tables as in a SQL database.

Mongoose is a popular npm package for interacting with MongoDB. With Mongoose, you can use plain JavaScript objects instead of JSON, which makes it easier to work with MongoDB. Also, it allows you to create blueprints for your documents called schemas, so you don't accidentally save the wrong type of data and cause bugs later.

In the MongoDB and Mongoose courses, you'll learn the fundamentals of working with persistent data including how to set up a model, and save, delete, and find documents in the database.

▼ Collapse course

12/12

- Install and Set Up Mongoose
- Create a Model
- Create and Save a Record of a Model
- Create Many Records with `model.create()`
- Use `model.find()` to Search Your Database
- Use `model.findOne()` to Return a Single Matching Document from Your Database
- Use `model.findById()` to Search Your Database By `_id`
- Perform Classic Updates by Running Find, Edit, then Save
- Perform New Updates on a Document Using `model.findOneAndUpdate()`
- Delete One Document Using `model.findByIdAndRemove`
- Delete Many Documents with `model.remove()`
- Chain Search Query Helpers to Narrow Search Results



HOLY ANGEL UNIVERSITY

SCHOOL OF COMPUTING

1st Semester, School Year 2025-2026



Back End Development and APIs Projects

Back End Development and APIs Projects

You've worked with APIs before, but now that you know npm, Node, Express, MongoDB, and Mongoose, it's time to build your own. Draw on everything you've learned up to this point to create 5 different microservices, which are smaller applications that are limited in scope.

After creating these, you'll have 5 cool microservice APIs you can show off to friends, family, and potential employers. Oh, and you'll have a shiny new Back End Development and APIs Certification, too.

Timestamp Microservice	✓
Request Header Parser Microservice	✓
URL Shortener Microservice	✓
Exercise Tracker	✓
File Metadata Microservice	✓

(view the timeline in my profile to see the source code)

Timeline for Finals Badging

Timeline		
Challenge	Solution	Completed
Back End Development and APIs Certification		Oct 2, 2025
File Metadata Microservice	View	Oct 1, 2025
Exercise Tracker	View	Oct 1, 2025
URL Shortener Microservice	View	Oct 1, 2025
Chain Search Query Helpers to Narrow Search Results		Oct 1, 2025
Delete Many Documents with model.remove()		Oct 1, 2025
Delete One Document Using model.findByIdAndRemove		Oct 1, 2025
Perform New Updates on a Document Using model.findOneAndUpdate()		Oct 1, 2025
Perform Classic Updates by Running Find, Edit, then Save		Oct 1, 2025
Use model.findById() to Search Your Database By _id		Oct 1, 2025
Use model.findOne() to Return a Single Matching Document from Your Database		Oct 1, 2025
Use model.find() to Search Your Database		Oct 1, 2025
Create Many Records with model.create()		Oct 1, 2025
Create and Save a Record of a Model		Oct 1, 2025
Create a Model		Oct 1, 2025

1 of 93

Timeline		
Challenge	Solution	Completed
Install and Set Up Mongoose		Oct 1, 2025
Legacy JavaScript Algorithms and Data Structures Certification		Oct 1, 2025
Cash Register	View	Oct 1, 2025
Telephone Number Validator	View	Oct 1, 2025
Roman Numeral Converter	View	Oct 1, 2025
Map the Debris		Sep 27, 2025
Make a Person		Sep 27, 2025
Build an Optional Arguments Sum Function		Sep 27, 2025
Build an All-True Property Validator		Sep 27, 2025
Binary Agents		Sep 27, 2025
Create a Deep Flattening Tool		Sep 27, 2025
Drop it		Sep 27, 2025
Smallest Common Multiple		Sep 27, 2025
Build a Sum All Primes Calculator		Sep 27, 2025
Build an Odd Fibonacci Sum Calculator		Sep 27, 2025

<< < 2 of 93 > >>



HOLY ANGEL UNIVERSITY

SCHOOL OF COMPUTING

1st Semester, School Year 2025-2026



Timeline

Challenge	Solution	Completed
Implement an HTML Entity Converter		Sep 27, 2025
Implement a Unique Sorted Union		Sep 27, 2025
Build a Missing Letter Detector		Sep 27, 2025
Implement a DNA Pair Generator		Sep 27, 2025
Build a Smart Word Replacement Function		Sep 27, 2025
Implement a Pig Latin Translator		Sep 27, 2025
Implement a Spinal Case Converter		Sep 27, 2025
Implement a Matching Object Filter		Sep 27, 2025
Implement a Value Remover Function		Sep 27, 2025
Diff Two Arrays		Sep 27, 2025
Design a Sum All Numbers Algorithm		Sep 27, 2025
Introduction to Currying and Partial Application		Sep 27, 2025
Use the some Method to Check that Any Elements in an Array Meet a Criteria		Sep 27, 2025
Use the every Method to Check that Every Element in an Array Meets a Criteria		Sep 27, 2025
Apply Functional Programming to Convert Strings to URL Slugs		Sep 27, 2025

<< < 3 of 93 > >>

Timeline

Challenge	Solution	Completed
Combine an Array into a String Using the join Method		Sep 27, 2025
Split a String into an Array Using the split Method		Sep 27, 2025
Return a Sorted Array Without Changing the Original Array		Sep 27, 2025
Sort an Array Alphabetically using the sort Method		Sep 27, 2025
Use Higher-Order Functions map, filter, or reduce to Solve a Complex Problem		Sep 27, 2025
Use the reduce Method to Analyze Data		Sep 27, 2025
Add Elements to the End of an Array Using concat Instead of push		Sep 27, 2025
Combine Two Arrays Using the concat Method		Sep 27, 2025
Remove Elements from an Array Using slice Instead of splice		Sep 27, 2025
Return Part of an Array Using the slice Method		Sep 27, 2025
Implement the filter Method on a Prototype		Sep 27, 2025
Use the filter Method to Extract Data from an Array		Sep 27, 2025
Implement map on a Prototype		Sep 27, 2025
Use the map Method to Extract Data from an Array		Sep 27, 2025
Refactor Global Variables Out of Functions		Sep 27, 2025

<< < 4 of 93 > >>

Timeline

Challenge	Solution	Completed
Pass Arguments to Avoid External Dependence in a Function		Sep 27, 2025
Avoid Mutations and Side Effects Using Functional Programming		Sep 27, 2025
Understand the Hazards of Using Imperative Code		Sep 27, 2025
Understand Functional Programming Terminology		Sep 27, 2025
Learn About Functional Programming		Sep 27, 2025
Implement the Chunky Monkey Algorithm		Sep 27, 2025
Implement the Mutations Algorithm		Sep 27, 2025
Where do I Belong		Sep 27, 2025
Implement a Falsy Remover		Sep 27, 2025
Implement the Slice and Splice Algorithm		Sep 27, 2025
Build a Title Case Converter		Sep 27, 2025
Build a Boolean Check Function		Sep 27, 2025
Build a First Element Finder		Sep 27, 2025
Implement the Truncate a String Algorithm		Sep 27, 2025
Build a String Repeating Function		Sep 27, 2025

<< < 3 of 93 > >>

Timeline

Challenge	Solution	Completed
Build a Confirm the Ending Tool		Sep 27, 2025
Build the Largest Number Finder		Sep 27, 2025
Build a Longest Word Finder App		Sep 27, 2025
Factorialize a Number		Sep 27, 2025
Build a String Inverter		Sep 27, 2025
Build a Celsius to Fahrenheit Converter		Sep 27, 2025
Request Header Parser Microservice	View	Aug 27, 2025
Timestamp Microservice	View	Aug 27, 2025
Caesars Cipher	View	Aug 27, 2025
Palindrome Checker	View	Aug 27, 2025
Use an IIFE to Create a Module		Aug 27, 2025
Understand the Immediately Invoked Function Expression (IIFE)		Aug 27, 2025
Use Closure to Protect Properties Within an Object from Being Modified Externally		Aug 27, 2025
Use a Mixin to Add Common Behavior Between Unrelated Objects		Aug 27, 2025
Override Inherited Methods		Aug 27, 2025

<< < 6 of 93 > >>



HOLY ANGEL UNIVERSITY
SCHOOL OF COMPUTING
1st Semester, School Year 2025-2026



Certificates

