BUILD A DEV

Build A Dev Syllabus

Welcome

Becoming a software engineer is one of the best career moves you can make. Build A Dev built this Nanodegree program with input from leaders in the software industry to provide world-class Full Stack Web Development instruction that will give you the tools needed to perform well in a variety of developer roles.

Program Features

1. **Instructor-led experience**

   You're not alone. Each class is limited to a maximum of 25 students and a dedicated instructor who is a passionate, industry veteran.

2. **Industry curated curriculum**

   Build confidence knowing course content was designed by industry-experienced developers based on the most in-demand technology stack..

3. **Unique classroom experience**

   Attend all your sessions in an e-learning environment, significant cost-savings we pass on to you

4. **Start building your coding portfolio**

   Learn better by doing. In addition to the weekly labs and techncial interview prep, you will be working on your own coding projects you can add to your portfolio

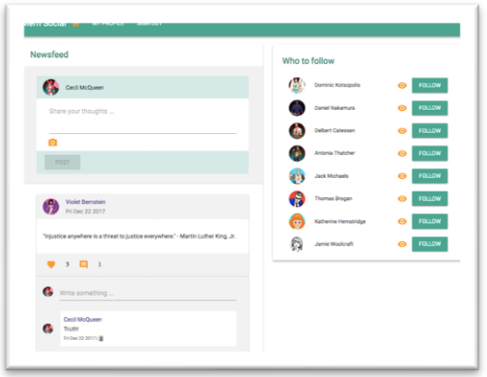
5. **Weekly interview prep**

   We believe in technical interviews demand weekly prep. They may seem intimidating initially, but like all things, the more you do it, the better you get.

6. **Career & job placement services**

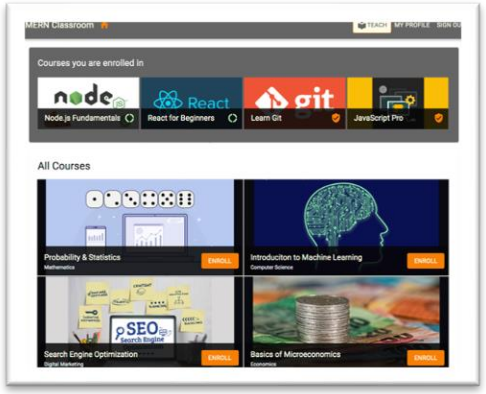
   Get 1:1 career coaching, access to an exclusive job board and career development course, and hands-on help with your resume and LinkedIn recommendations.

Projects



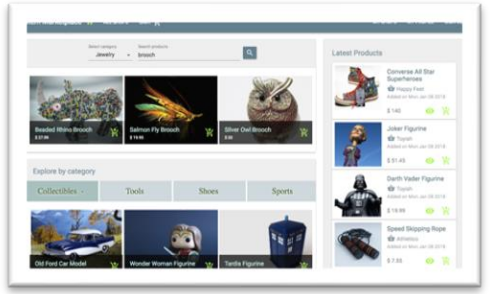
Social Media Platform

The social media platform will implement simple features such as post sharing, liking and commenting, following friends, and an aggravated news feed.



Learning Management System

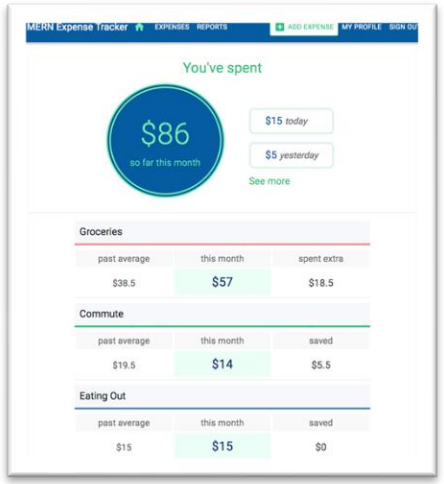
This application will implement features that allow instructors to add courses with lessons, while students can enroll in these courses and track their progress



Online Marketplace

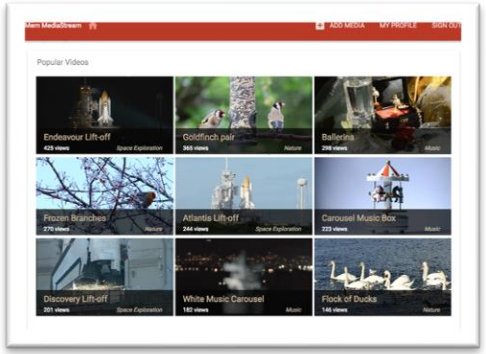
The features of this marketplace will cover aspects such as support for seller accounts, product listings, a shopping cart for customers, payment processing, order management, and real-time bidding capabilities.

Projects



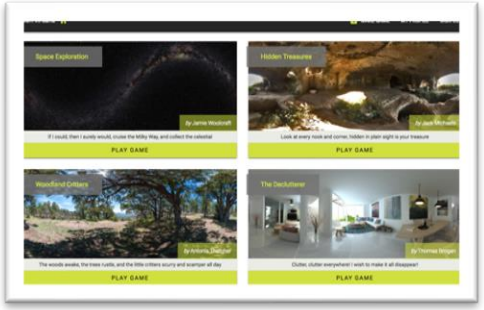
Expense Tracker

This application will add the expenses incurred over time. Then, the application will extract data patterns to give the users a visual representation of how their expense habits fare as time progresses.



Media Streaming Application

We will implement content uploading and viewing capabilities with a media content upload feature for content providers, and real-time content streaming for viewers.



Web VR Games

We will explore how to create rare web experiences with react 360 in MERN by putting together a basic VR game application for the web

HTML

1. Intro to HTML & CSS

- What is HTML?
- What is CSS?
- How HTML & CSS work together.

2. HTML Syntax

- How HTML tag work
- Adding Comments
- Structuring an HTML tag
- The <meta> tag
- What is semantic markup?

3. Basic Elements

- HTML Text Formatting
- Paragraphs and Headings
- Lists
- Quoting a block of Text
- Formatting Text Inline
- Marking Up Code

4. Links

- Link Markup
- URL Structure
- Internal Vs. External Linking
- Relative Vs. Absolute Linking
- Other Types of Links
- Link Targets

5. Structure & Layout with HTML

- Webpage Layout
- Block vs Inline Elements
- Page Sections
- Building a Blog Article Layout

6. Media

- Images
- Adding Images to a Page
- Responsive Images: Considering Different Devices
- The <picture> Element
- Using SVG
- Other Media
- Embedding Video
- Embedding Audio
- Storing Multimedia Files

7. Tables & Structured Data Elements

- Tables
- Description Lists
- Why is Structured Data Important

8. Web Forms

- Interacting with Webpages
- Components of an HTML Form
- The <form> Element
- Form Fields
- Labelling Fields
- Setting Up a Basic Form
- Creating Select Boxes
- Creating Radio Buttons
- Creating Checkboxes
- Creating Email Forms
- Special Field Types
- The <meter> Element
- Validating Forms

CSS

1. Intro to CSS

- What are Styles?
- What does Cascading Mean?
- CSS Syntax
- Using CSS in your Webpage
- External Style Sheets
- Commenting Your CSS Code

2. Targeting Elements

- Targeting Elements by Tag
- Targeting Elements by Class
- The Cascade, Inheritance and Parent-Child Relationships.
- Selecting Elements by their Relationships
- Specify and Precedence in the Cascade
- Targeting Elements with Specific Attributes
- Advanced Targeting

3. Styling Text

- Choosing Fonts
- Google Fonts
- Including External Fonts with @font-face
- Sizing Text
- Formatting for Readability

4. Color in CSS

- How Computer Monitor Work
- Representing Color in CSS
- Gradients
- The border Property

CSS

5. Page Layout with CSS

- The Box Model
- Padding and Margins
- Element Flow
- Creating Layers and Overlapping Elements
- Creating an Overlay Using z-index.

6. CSS Grid & Flexbox

- Modern Solutions
- Using Flexbox
- Using CSS Grid Layout
- Browser Support

7. Responsive Design & Media Queries

- Defining Media Queries
- Responsive Layouts
- Making a Full-width Layout Responsive
- Not Just For Screen Widths

8. CSS Transformations &Animations

- CSS Transitions
- CSS Transformation
- CSS Animations

9. CSS Variables

- What Are Variables?
- Simplifying Styles with Variables
- Calculations with Variables

10. CSS Preprocessors

- How CSS Preprocessors Work
- Getting Started with Sass
- Writing Sass.

11. Website Performance

- What does Performance entail
- Performance testing Tools.
- Minify HTML and CSS Files
- Optimize Your Images
- Load Your Critical CSS FIRST

JAVASCRIPT

1. Intro to JavaScript

- Learn what you can do with JavaScript
- Take a look at JavaScript's features
- Learn the pros and cons of JavaScript.
- Running code using a REPL
- Learn basic JavaScript syntax

2. JavaScript Data Types

- Learn about JavaScript's "Number" type.
- Learn about JavaScript's "String" type.
- Learn about JavaScript's "Boolean" type.
- Learn about JavaScript's "Object" type.
- Learn about arrays in JavaScript
- Learn about JavaScript's "Function" type.
- Learn about JavaScript's "Undefined" type.
- Learn about JavaScript's "BigInt" type.
- Learn about JavaScript's Symbol" type.

3. JavaScript Control Flow

- Learn about equality in JavaScript.
- Learn about if statements in JavaScript.
- Learn about For Loops in JavaScript.(For-in and For-of)
- Learn about While Loops in JavaScript.(while and do-while)
- Handling and throwing errors in JavaScript
- Learn about switch case in JavaScript.
- Learn about ternary operators in JavaScript.
- Learn about the difference between block-scoped & function-scoped variables

4. OOP in JavaScript

- Learn about JavaScript classes
- Learn about Prototype-Based Inheritance
- Using "This" keyword in JavaScript.

JAVASCRIPT

5. Objects, Arrays & Functions in JS

- Using built-in Object Functions
- Using built-in Array Functions

6. JavaScript ES6+ Syntax

- Write functions using Arrow syntax
- Set default values for function arguments
- Learn about the “spread” operator
- Use object destructuring to get object properties
- Use Interpolation to insert data intro strings
- Learn the Import & Export code in JavaScript

7. Write & Run a JavaScript Program

- Learn about writing & running JavaScript programs
- Write a JavaScript program for the browser
- Move JavaScript to an external script
- Install Node.js and NPM
- Write a Node script
- Use Babel to transpile code

8. Asynchronous JavaScript

- Use callbacks to handle asynchronous operations
- Learn about Promises
- Use Async and Await to make asynchronous code readable.

9. Create a Web Server with JavaScript

- Learn the basics of writing s Node server.
- Create and set up a Node.js project
- Create and run a basic Express server
- Create and test a GET endpoint
- Read a file with the ‘fs’ package
- Create and test a POST endpoint

10. JavaScript Best Practices

- Use Eslint to ensure code styles
- Ensure Immutability in JavaScript programs
- Avoid Loops when working with arrays

MERN

1. Foundations

- Learn the basics of how React applications can be built
- Compile React JSX at runtime & them move the compilation as well as serving the files to the server
- Installing nvm(Node Version Manager) & using NPM to install Node.js packages & save command-line instructions in easy-to-spot scripts
- Use Babel to compile from one specification of the language to another
- Get a whiff of what Node.js & Express can do
- Gain familiarity with using git & comparing your code with the testable code on Github & understanding the diffs between each step.

2. React Components

- Create React Classes & instantiate components
- Writing fine-grained individual components & composing them in an enclosing component
- Pass parameters or data from an enclosing component to its children
- Reusing a component class and rendering it differently with different data
- Dynamically using a map() to generate components based on an array of input data
- Pass data among components & create components dynamically from data

3. React State

- Learn how to use state & make changes to it on user interactions or other events
- Learn hot state values are propagated down the hierarchy as props
- Learn how a child can communicate with its parent via callbacks
- Use simulated asynchronous calls and data local to the browser to achieve all this

4. Express and GraphQl

- Compare two API standards: REST & GraphQl
- Using GraphQL, learn how to build the C & R part of CRUD
- Learn how to implement validations
- Learn about GraphQL strong type system and its benifits
- Learn how to persist data: work with server's memory instead of the browser memory

MERN

5. MongoDB

- Learn about MongoDB, documents & collections
- Learn about the installation and other ways of getting access to an instance of a database in MongoDB
- Install or use MongoDB on the cloud
- How to use the mongo shell and the Node.js driver to access the basic operations in MongoDB the CRUD operations
- Modify server code to replace APUI calls to read & write from a MongoDB database instead of an in-memory array.

6. Architecture & ESLint

- UI Server
- Multiple Environments
- Proxy Based Architecture
- ESLint
- ESLint For Front End
- React Prototypes

7. Modularization and Webpack

- Backend Modules
- Front End Modules and Webpack
- Transform and Bundle
- Libraries Bundle
- Hot Module Replacement
- Debugging

8. React Router

- Simple Routing
- Route Parameters
- Query Parameters
- Links: Full Source | Diffs
- Programmatic Navigation
- Nested Routes
- Browser History Router

MERN

9. React Forms

- Controlled Components
- Controlled Components in Forms
- More Filters
- Types Inputs
- Edit Form
- Number Input
- Date Input
- Text Input
- Update API
- Updating an Issue
- Updating a Field
- Delete API
- Deleting an Issue

10. React-Bootstrap

- Bootstrap Installation
- Buttons
- Navigation Bar
- Panels
- Tables
- Forms
- The Grid System
- Inline Forms
- Horizontal Forms
- Validation Alerts
- Toasts
- Modals

11. Server Rendering

- New Directory Structure
- Basic Server Rendering
- Webpack for the Server
- HMR for the Server
- Server Router
- Hydrate
- Data from APIs
- Syncing Initial Data
- Common Data Fetcher
- Generated Routes
- Data Fetcher with Parameters
- Data Fetcher with Search
- Nested Components
- Redirects

12. Advanced Features

- Higher Order Component for Toast
- MongoDB Aggregate
- Issue Counts API
- Report Page
- List API with Pagination
- Pagination UI
- Undo Delete API
- Undo Delete UI
- Text Index API
- Search Bar

MERN

13. Authentication

- Sign-In UI
- Google Sign-In
- Verifying the Google Token
- JSON Web Tokens
- Signing Out
- Authorization
- Authorization- Aware UI
- React Context
- CORS with credentials
- Server Rendering with Credentials
- Cookie Domain

14. Writing Tests

- Testing with Mocha
- Our first Mocha test
- Starting the back end with Mocha
- Verifying the correct routing
- Testing GraphQL with Mocha
- Testing the authentication
- Testing authenticated requests
- Testing React with Enzyme

15. Deployment: CircleCL & Heroku

- Preparing the final production build
- Code-splitting with React Loadable and webpack
- Code-splitting wwith SSR
- Setting up Docker
- What is Docker?
- Installing Docker
- Dockerizing your application
- Writing your first Dockerfile
- Building and running Docker containers
- Multi-stage Docker production builds
- Amazon Relational Database Service
- Configuring Continuous Integration
- Deploying applications to Heroku

www.buildadev.com

BUILD A DEV