To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Full-Stack Web Development with Vue and FastAPI

Class Duration

5 days

Target Audience

All students must have Python, JavaScript, and HTML programming experience. Experience with CSS is helpful, but not required.

Description

This comprehensive course covers modern Vue 3 with TypeScript and its integration with FastAPI, a high-performance web framework for building APIs with Python. The course starts with the fundamentals of Vue, including how to set up a development environment with Vite, build single-file components with the Composition API and `<script setup>`, and handle events, forms, state management with Pinia, and routing with Vue Router 5. It then dives into the basics of HTTP and URLs, databases with async SQLAlchemy, and Python web frameworks. The course concludes with a detailed guide on how to build a REST API with FastAPI — including Pydantic models, async routes, cookies, errors, and specific status codes — how to secure a REST API with JWT and configure CORS, how to connect the Vue front end to the API, and how to deploy the full stack. This course is perfect for professionals who want to build robust, full-stack applications using Vue and FastAPI.

Objectives

- Understand the fundamentals of Vue 3 and how it compares to other frameworks.
- Set up a Vue development environment with `create-vue`, Vite, and TypeScript.
- Build single-file components using the Composition API with `<script setup>`.
- Understand Vue's reactivity system (`ref`, `computed`, `watch`) and component communication with props, emits, and slots.
- Learn about event handling, data binding, and forms in Vue.
- Manage application state with Pinia and client-side routing with Vue Router 5.

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Fetch and render asynchronous data from a REST API with loading and error states.
- Understand the basics of HTTP and URLs, databases, and Python web frameworks.
- Learn how to build a REST API with FastAPI, including Pydantic models, async routes, cookies, errors, and specific status codes.
- Access a database asynchronously with SQLAlchemy from FastAPI routes.
- Understand how to secure a REST API with JWT and configure CORS.
- Connect a Vue front end to a FastAPI backend and deploy the full stack.

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Student will need permission to install Python, Node.js and Visual Studio Code on their computers. Also, students will need permission to install Conda/Pip Packages, NPM Packages, and Visual Studio Extensions. If students are unable to configure a local environment, a cloud-based environment can be provided.

Training Topics

Introduction to Vue 3

- What is Vue? What problem does it solve?
- Vue as a progressive framework
- Composition API vs. Options API: what changed and why
- Vue vs. React, Svelte, and Angular reactivity models

Development Environment

- Requirements
- Project scaffolding with `create-vue`
- Vite dev server and HMR

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- Vue extension for Visual Studio Code
- Run/Debug a Vue app in Visual Studio Code
- Vue support in WebStorm
- Vue DevTools

Single-File Components

- Anatomy of a `.vue` file: `<template>`, `<script setup>`, `<style>`
- Scoped CSS
- HTML, CSS, and TypeScript content
- How SFCs are compiled

Template Syntax

- Text interpolation and expressions
- Attribute binding with `v-bind` (`:`)
- Conditional rendering (`v-if`, `v-else-if`, `v-else`, `v-show`)
- List rendering with `v-for` and `key`
- Class and style bindings

Reactivity Fundamentals

- `ref()` for reactive values
- `reactive()` for objects — and when to prefer `ref`
- `computed()` for derived values
- `watch()` and `watchEffect()` for side effects
- Updating arrays and objects with deep reactivity
- Reading legacy Options API code

Component Basics

- Component file structure
- Props with `defineProps()` and TypeScript types
- Events with `defineEmits()`
- Two-way component binding with `defineModel()`

Component Composition

- Nested components
- Slots: default, named, and scoped
- Provide/inject for cross-tree data
- Composables: extracting reusable logic
- Component tree best practices

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Event Handling

- Listening with `v-on` (`@`)
- Event modifiers (`.prevent`, `.stop`, `.once`)
- Key and mouse modifiers
- Emitting and handling component events

Data Binding and Forms

- Two-way binding with `v-model`
- `v-model` on inputs, selects, checkboxes, and radios
- `v-model` modifiers (`.lazy`, `.number`, `.trim`)
- Form validation and submission patterns

Lifecycle and Effects

- `onMounted`, `onUpdated`, `onUnmounted`
- Watcher cleanup functions
- `nextTick()` for awaiting DOM updates
- Template refs for DOM access

State Management with Pinia

- Why Pinia? (the official store, replacing Vuex)
- Setup stores: state, getters, and actions
- `storeToRefs()` and preserving reactivity
- Sharing stores across components

Routing with Vue Router 5

- What is routing? What problem does it solve?
- Routes, dynamic params, and nested routes
- `<RouterLink>` and `<RouterView>`
- Programmatic navigation with `useRouter` and `useRoute`
- Navigation guards
- File-based routing in Vue Router 5

Asynchronous Data

- Fetching data with `fetch` and composables
- Loading and error states
- Async components and `<Suspense>` (experimental) overview

Quick Review of HTTP and URLs

- What is HTTP?

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved

- What are HTTP Verbs?
- What are the parts of a URL?
- Path Parameters
- Query String Parameters
- Common HTTP Headers
- Common HTTP Status Codes

Quick Review of Databases

- What is a Database?
- Connecting to a Database
- Query data from a Database
- Modify data in a Database

Overview of Python Web Frameworks

- Django
- Flask
- FastAPI

Representational State Transfer

- What is REST?
- HTTP Verbs and their use in REST
- OpenAPI
- JSON
- Swagger

Building a REST API with FastAPI

- Create an Application
- Define a Hello World Route
- Configure a Path Parameter
- Configure a Query String Parameter
- Reading Request Body with Pydantic Models
- Writing a Response Body with Response Models
- Validation with Pydantic
- Handling Cookies
- Handle Errors
- Return Specific Status Codes
- Asynchronous Routes
- Serving Static Files

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Database Access with Async SQLAlchemy

- SQLAlchemy async engine and sessions
- Defining models
- Dependency injection of database sessions
- CRUD operations from FastAPI routes

Middleware

Security

- Secure a REST API with JWT
- Configure CORS

Connecting Vue to FastAPI

- Calling the API from Vue composables
- Sharing API state with Pinia
- Handling auth tokens in the front end
- Login flows and route guards

Deployment

- Building the Vue app for production with Vite
- Serving the front end and API together
- Containerizing the full stack

Conclusion

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved