CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

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

Desktop Apps with Rust and Tauri

Duration

3 days

Description

Many programmers think of Rust as only a systems programming language, but it can be so much more. Rust is a great language for building desktop applications using the architectural pattern popularized by Electron which combines a web browser app and a server app to form a cross-platform desktop app. The Rust framework that enables this is combination is called Tauri and unlike Electron, it is extremely fast and more memory effcient. In this course, you'll learn how to build desktop applications with Rust and Tauri. You'll learn how to create UI elements such as windows, menus, and tray icons. Also, you'll learn how to create dialogs, notifications, and file system dialogs. We will go deeper and learn how to create file system watchers and how to interact with the operating system. By the end of this course, you'll be able to create your own desktop applications with Rust and Tauri. For the web browser app part of the application, this course can be taught with Leptos, React, Angular, Svelte or Solid.js. Your choice.

Objectives

- Modern Approaches to Desktop App Development
- Explore how to Create Desktop Apps with Rust and a Web Browser
- Create a new Desktop App using Tauri and a Browser-based Frontend
- Learn how to integrate popular JS frameworks or Rust WASM frameworks with Tauri
- Create UI elements such as Windows, Menus, and Tray Icons
- Connect a Tauri App to a Database (PostgreSQL or SQL Server)
- Connect a Tauri App to a REST API
- Learn how to Publish and Distribute a Tauri App

Prerequisites

- This course assumes prior experience with Rust, JavaScript, HTML, and CSS

CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

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

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. Students will need permission to install Rust and Visual Studio Code on their computers. Also, students will need permission to install Rust Crates and Visual Studio Extensions. Students will need a local instance of Postgresql or SQL Server installed on their computer (using Docker is acceptable). If students are unable to configure a local development environment on their machines, a cloud-based environment can be provided.

Outline

- Introduction
- Desktop Apps with Tauri
  - What is Tauri?
  - How does it compare to Electron, Qt, and WinForms?
- Getting Started
  - Create a Tauri Project
  - Integrate with JavaScript or Rust-based browser UI framework
  - Run and Debug Tauri Applications
- Call Backend Rust Code from the Front-End
  - Define Rust Backend Functions
  - Call Rust Backend Functions
  - Pass Arguments to Rust Backend Functions
  - Return Data from Rust Backend Functions
  - Handle Errors from Rust Backend Functions
  - Asynchronous Rust Backend Functions
- Interact with App Parts from the Rust Backend
  - Window
  - AppHandle
  - Managed State
  - Multiple Commands
- UI Elements
  - Window

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

  - Menu
  - Tray Icon
  - Splash Screen
  - Dialog
  - Notification
- Interacting with the Operating System
  - Create a File System Dialog
  - Create a File System Watcher
- Database Programming
  - What is a Database?
  - Connect to Postgresql
  - Query data from the database
  - Modify data in the database
  - Connect Tauri to a Database
- Deployment
  - Create Platform Specific Installers
  - Cross-Platform Compilation with CI/CD
  - Signing the Application
- Conclusion