

**Developing Web Applications with Flask**

**Course Number:** PYTH-140
**Duration:** 3 days

**Overview**

Accelebrate's Flask Training: Developing Web Applications with Flask course teaches Python developers to create professional web applications with the Flask microframework.

**Prerequisites**

All attendees must have basic Python programming skills and an understanding of HTML fundamentals.

**Materials**

All attendees receive comprehensive courseware and a textbook.

**Software Needed on Each Student PC**

* Any Windows, Linux, or macOS operating system
* Python 3.x installed (Anaconda bundle recommended)
* The Flask Library
* A text editor or IDE (PyCharm Community Edition recommended)

**Objectives**

* Create complete Flask applications
* Work with session data
* Design Jinja templates using inheritance
* Integrate a SQLite database
* Test and debug Flask applications

**Outline**

* Introduction
* An Overview of Flask
	+ Origins and purpose
	+ Views, controllers, but no models
	+ What Flask can and cannot do
* Diving Right In
	+ A simple application
	+ Configuring routes
	+ Defining view functions
	+ Using templates
	+ Deploying the application
* Simple Views
	+ HTTP Methods
	+ Returning content
	+ Templates and variables
* Organizing the Application
	+ Configuring routes
	+ Static files
	+ Configuration
	+ Context locals
	+ Sessions
* Integrating a SQLite Database
	+ The g (global) object
	+ Creating a table
	+ Inserting rows
	+ Making queries
	+ Walkthrough of minor changes necessary for integrating other databases (SQL Server, Oracle, MySQL, etc.)
* Jinja Templates
	+ The big picture
	+ Simple syntax
	+ Block statements
	+ Loading and rendering
	+ Inheritance
	+ Advanced syntax
	+ Filters and tests
* HTTP Responses
	+ String responses
	+ Setting the status code
	+ Getting the response object
	+ Adding response headers
* HTML Forms
	+ Simple forms
	+ Form submission
	+ Retrieving user data
	+ Re-submitting forms
* Advanced Views
	+ Redirects
	+ Auto-escaping HTML
	+ Message flashing
	+ Built-in decorators
* Error Handling
	+ Handling an unrouted URL
	+ Aborting a page view
	+ Customizing the error page
* Testing and Debugging
	+ Using unittest
	+ Simple tests
	+ Accessing sessions
	+ Mocking resources
* Advanced Topics
	+ Custom filters and tests
	+ Class-based views
	+ Support for JSON
	+ Blueprints
* Conclusion