# Python for Web Development

## ABOUT DJANGO FRAMEWORK

### What are the frameworks availabe in python?

- Django
- Pyramid
- Flask
- Web2py
- Tornado
- Pylons
- Bottle
- Numpy

### Why do we choose Django  over flask?

### Reason to choose Django

- It is fast and simple
- It is secure
- It suits for any web application projects
- Reason to choose Flask
- Extremely flexible
- Minimalist without sacrificing power
- Simple to learn and use
- Routing URLs is easy
- Small core and easily extensible

# Course Content

## CHAPTER 1: INTRODUCTION

- Introduction to Django
- What is Django?
- Django and Python
- MVT
- Pypi download and install Django

## CHAPTER 2: Front-End Introduction & HTML levels

- What is the Web?
- HTML Level One Introduction
    - Basics
    - Basic Tagging
    - Lists
    - Divs and Spans
    - Attributes
- HTML Level One Assessment -Overview
- HTML Level Two – Advanced
    - Level Two Introduction
    - Tables
    - Forms
    - Labels
    - selections
- HTML Level Two –Assessment

## CHAPTER 3: CSS ,BOOTSTRAP AND JAVASCRIPT

- CSS basics
    - Fonts
    - Box models
    - Spectrum project overview
- Bootstrap overview
    - Introduction
    - Buttons
    - Forms
    - Navbars

- Grid
- Bootstrap Assessment

- Javascript Overview
  - Introduction
  - Basics
  - Connecting javascript
  - Operators
  - Control flow
  - Loops concepts
  - Functions
  - Array
  - Object
  - Javascript Assessment

## CHAPTER 4: Document object model, jquery and Back end intoduction

- Document object model
  - Introduction
  - DOM Interaction
  - Content Interaction
  - Events
  - Game Project overview and solutions
- Jquery
  - Introduction
  - Basics
  - Event
  - Project overview and solutions
- Backend introduction
  - Introduction
  - Command line and Terminal overview

## CHAPTER 5: Python

- Introduction
- Installation and setup
- Numbers
- Strings
- Lists
- Dictionaries
- Tuples, sets and Booleans

- Control flow
- Function
- Oops concept
- Regular expressions
- Modules and Package
- Decorators
- Name and main
- Lab Session 5

## CHAPTER 6: DJANGO

- Django setup
- Challenge Task and solutions
- URL Mappings
- Templates
- Static files
- Creating models
- Population Scripts
- Models-Templates-Views Paradigm
- Django Forms
- Forms basics code along
- Form validation
- Model forms
- Relative URLs with Templates
- URL Template Inheritance
- Template Filters and Custom Filters
- Django password
- User Models
- Coding User Models and Forms
- Registration
- Logins
- Lab session 6

## CHAPTER 7: Django Deployment

- Introduction
- Setting up GitHub
- Full Deployment Walkthrough on Python Anywhere

## CHAPTER 8: Advanced Topics - CBVs

- Introduction
- Hello World with CBVs
- Template Views with CBV
- Detail View and List View
- CRUD Views
- Introduction to Debug Toolbar
- Debug Toolbar Overview
- Lab Session 8

## CHAPTER 9: Advanced Topics - Customizing the Django Admin

- Introduction to Admin
- Admin Templates
- Ordering Fields
- Adding Search
- Adding Filters
- Adding Fields
- Editable List View
- Lab Session 9

# About Rest Framework

Django REST framework is a great and flexible framework for building Web APIs.

Some of the reasons you might want to use Django REST framework:

- The Web browsable API is a vast usability win for your developers.
- Validation policies including packages for OAuth1a and OAuth2.
- One-off that supports both ORM and non-ORM data sources.
- Customizable all the method down - just use regular function-based views if you don't want the more powerful features.
- Wide documentation, and countless community support.
- Used and reliable by globally recognized companies including  Red Hat, Heroku,etc.,

# Rest Framework Course content

## CHAPTER 1: INTRODUCTION

- What is REST?
- Why REST
- What and Why DJango REST Framework
- DRF setup
- RESTful Structure
- DRF Quick start
    - Model Serializer
    - Update views
    - Update URLs
    - Test
- Refactor for REST
    - GET
    - Datetime format
    - Post
    - Author Format
    - Delete

## CHAPTER 2: REST In Action

- Creating the Project
- Create a view
- Configure the URL and Test
- Create APP Level URLs
- Create a model Class
- Configure the database and run migrations
- Using the models, and show in views and test
- Model questions

## CHAPTER 3: Function based views and Serializers

- DRF Components
- Function Based views
- Serializers
- Create the project
- Create the model
- Create the Serializer
- GET single student

- Create Student
- Implement Non primary key based Operations
- Use @API_VIEW
- Configure the URLs
- Test and Test using Postman
- Function based views and serializers
- Lap Pratice 3

## CHAPTER 4: Class based views and Mixins

- Introduction
- Creating the Project
- Implementing primary and nonnprimay key based operations
- Configure the URLs and Test
- Mixins in primary and non primary Key based operations
- Lab CHAPTER 4

## CHAPTER 5: Generic views and viewsets

- Introduction
- Genric in action
- Create voewsets
- Configure URLs and TEST
- Lab CHAPTER 5

## CHAPTER 5: Nested Serializers

- Create the project
- Create the model
- Create the serializers
- Create the REST endpoints
- Congigure the URLs

## CHAPTER 6: Pagination and Security

- Pagination in action
- Pagination at class level
- Using LimitOFFsetPagination
- Authentication in action
- Authorization in Action
- Global Security

## CHAPTER 6: Flight Reservation in API

- Usecase
- Create the Project
- Create the model class
- Create the Reservation model
- Create the Serializers
- Create the Viewsets
- Configure the Router
- Run migrations
- Initial round of testing
- Implement findflights endpoint
- Test Findflight
- Implement Save Reservation
- Test Save Reservation