REPORT – 2

PYTHON WEB FRAMEWORKS

**Web framework -** collection of packages or modules which allow developers to write Web applications without having to handle low-level details as protocols, sockets or thread management

**Uses -**  makes it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application

**Advantages -**

* Open-source
* Good Documentation
* Efficient
* Secure
* Integration

**Few Operations Involved -**

* **Url Routing -** Routing is the mechanism of mapping the URL directly to the code that creates the web page.
* **Input form handling and validation -** Suppose you have a form which takes some input, the idea is to validate the data and then save it.
* **Output formats with template engine -** A template engine allows the developers to generate desired content types like HTML, XML, JSON.
* **Database connection -** Database connection configuration and persistent data manipulation through an ORM.
* **Web security -** Frameworks give web security against cross-site request forgery aka CSRF, sql injection, cross-site scripting and other common malicious attacks.
* **Session storage and retrieval -** Data stored in the session storage gets cleared when the page session ends.

**Types of Python Frameworks -**

**1. Full-Stack Framework**

One-stop-solution for all developer requirements. Form generators, form validation, and template layouts are usually available with a typical full-stack framework.

**2. Microframework**

Lightweight frameworks that don’t offer additional functionalities and features, such as database abstraction layer, form validation, and specific tools and libraries. Developers using a microframework need to add a lot of code and additional requirements manually.

**3. Asynchronous Framework**

Gaining popularity recently, any asynchronous framework is a microframework that allows for handling a large set of concurrent connections. Typically, an asynchronous framework built for Python uses the programming language’s asyncio library.

**Full-Stack Framework -**

**Django**

It is a free-to-use and open-source full-stack Python framework including a grand number of built-in features rather than offering them as individual libraries. Django makes use of its ORM for mapping objects to database tables.

This results in allowing the code to work across different databases as well as making it easier to migrate from one database to the other. Though Django has inherent support for MySQL, PostgreSQL, SQLite, and Oracle Database, it can support other databases via third-party drivers.

Key highlights -

* A plethora of ready-to-use libraries
* Authentication support
* Database schema migrations
* Object-relational mapper (ORM)
* Support for web servers
* Template engine
* URL routing

**WEB2PY**

The open-source Python framework comes with its own web-based IDE, which includes a code editor, debugger, and one-click deployment.

Though Web2Py allow users to create dynamic web content in Python, it doesn’t provide support for Python 3. The ticketing system is one of the most important features of the Python framework. The system issues a ticket to the user whenever an error occurs.

Key highlights -

* Ability to run on any web hosting platform that provides support for either Python or Java and Python
* Backward compatibility
* Built-in data security for preventing several common vulnerabilities, including cross-site scripting, injection flaws, and malicious file execution
* Devoid of installation and configuration requirements
* Follows MVC-pattern
* Provides support for internationalization
* Readability of multiple protocols
* Role-based access control

**Microframework -**

**Flask**

Because of its lightweight and modular design, Flask is readily adaptable.It allows the developers to build a solid web application foundation from where it is possible to use any kind of extensions required.It is compatible with Google App Engine.

**Key highlights -**

* Built-in fast debugger
* HTTP request handling
* Inbuilt development server
* Jinja2 templating
* RESTful request dispatching
* Support for plugging in any ORM
* Supports secure cookies to establish client-side sessions
* Unicode-based
* Unit testing support
* WSGI 1.0 compliance

**Falcon**

Aimed at rapidly building web APIs, Falcon is another widely used Python framework. Unlike other Python frameworks that require loading a lot of dependencies for building HTTP APIs, Falcon allows developers to build a cleaner design that enables HTTP and REST architectures.

Falcon is able to handle most requests with the same hardware than all its contemporaries.

**Key highlights -**

* An extensible, highly-optimized code base
* DRY request processing through middleware components and hooks
* Ease of access for headers and bodies via request and response classes
* Extra speed boost with Cython support
* Idiomatic HTTP error responses
* REST-inspired resource classes and URI templates offer intuitive routing
* Unit testing via WSGI helpers and mocks
* Upfront exception handling

**Asynchronous Framework**

**AIOHTTP**

It relies heavily on Python 3.5+ features, such as async & awaits.It makes use of Python’s asyncio library.In addition to being a server web framework, AIOHTTP can also serve as a client framework. It provides a request object and router to enable the redirection of queries to functions developed to handle the same.

Key highlights -

* Allows effectively building the views
* Middlewares support
* Pluggable routing
* Signals
* Supports both Client WebSockets and Server WebSockets without the Callback Hell

**TORNADO**

It is open-source Python framework and an asynchronous networking library. In addition to solving the C10k issue (which simply means to handle 10k connections at any given time), the asynchronous framework uses a non-blocking network I/O.

The Tornado is an ideal tool for building apps asking for high performance and several thousand concurrent users.

**Key highlights -**

* Allows implementation of 3rd-party authentication and authorization schemes
* Offers high-quality output
* Real-time services
* Supports translation and localization
* User authentication support
* Web templating