

Python as a Tool for Web Server Application Development

Sheetal Taneja¹, Pratibha R. Gupta²

ABSTRACT

With evolution of web, several competitive languages such as Java, PHP, Python, Ruby are catching the attention of the developers. Recently Python has emerged as a popular and the preferred web programming language, because of its simplicity to code and ease of learning. Being a flexible language, it offers fast development of web based applications. It offers development using CGI and WSGI. Web development in Python is aided by the powerful frameworks such as Django, web2py, Pyramid, and Flask that Python supports. Thus Python promises to emerge as one of the preferred choice language for web applications.

KEYWORDS

Python, Web server application development, frameworks, WSGI, CGI, PHP, Ruby

1. INTRODUCTION

Web is a rapidly growing repository of resources. Internet is used as a medium for accessing these resources. Web architecture mainly comprises of two entities, namely client and server [1]. Web client is an application (browser) on host machine that urges these resources, and web server is a machine on web that is responsible for fulfilling the request issued by client. Hypertext Transfer Protocol (HTTP) is the most popular protocol used by client and server for web communication. In a static web, browser issues HTTP request to the HTTP server, which searches for the requisite resource in its database and returns it as an HTTP response. To avoid any compatibility issues, every request issued by browser is in form of a URL (Uniform Resource Locator). The URL protocol defines the rules for communication between client and server. It comprise of host name (IP address) which helps in identifying the server system on the web, port number which determines the service (for example, FTP, email service) on the server that should respond to request, and the access path of the resource (web page) on server. The web where responses are already stored in server database in form of static web pages is termed static web. However, response returned by server to the client may be generated on the fly depending upon the request of the client. For example, a query provided to google search engine will dynamically generate web page of indexed results. Such a web is termed dynamic web. Since for generating the reply, server needs to process the user-

specific request, there is a program running at the server end that performs this task. Such a program is termed Web application [2]. Thus, web applications are the programs running at application server and are invoked using browser through the Internet.

Web applications offer several benefits over traditional applications that are required to be installed at each host computer that wishes to use them [3]. Web applications do not incur publishing and distribution costs as opposed to traditional applications where the software (applications) were published using CD's and distributed. They need not be installed at each client host; rather they are placed at a central server and accessed by large number of clients. Since a web application is managed centrally, application updates and data backups can be performed easily. The web applications are easily accessible irrespective of the boundaries of space and time. Since, they are accessed through browser, the particular platform accessing them is not an issue, and thus they provide cross-platform compatibility. In spite of above mentioned advantages, web applications have a few limitations. Internet connectivity and server availability is required for accessing web application through browser. However, accessing them through Internet may take more time as compared to applications installed on host systems. Also, web applications require compatible web browsers. Since they are deployed on web, they are vulnerable to several Internet attacks.

Python has found its use in numerous applications because of its simplicity. It has now emerged as the popular web programming language because of the functionalities offered by it. Being a flexible language, it is gaining the attention of web developers. Python eases the task of web programming by offering web frameworks which helps in fast web development where developers need not program from the scratch [4]. While some of these frameworks provide support for every component building, others provide minimal requisite functionalities.

Since the development of Internet, web evolved with several web applications catering to the needs of large number of users. Some commonly used web applications are webmail, Wikipedia, Google Apps, and Microsoft Office Live. Several big firms such as Google, Microsoft, Facebook, Twitter, Youtube, and Amazon depend heavily on web applications. Web application architecture is three-tier architecture [5] as shown in Figure 1. First tier is the client

¹Dyal Singh College, Computer Science Department, University of Delhi

²Sri Aurobindo College, Computer Science Department, University of Delhi

Email: ¹sheetaltaneja1@gmail.com, ²gpratibha11@gmail.com

Copyright ©IJICCT, Vol II, Issue I (Jan-Jun2014): ISSN 2347-7202

which is a web browser which renders the static/dynamic content returned by server, middle tier is the server which uses dynamic web languages and tools such as Python, PHP, Ruby, ASP.NET for web application development which handles the user specific request, and returns the data. Third tier is the database which provides storage means.

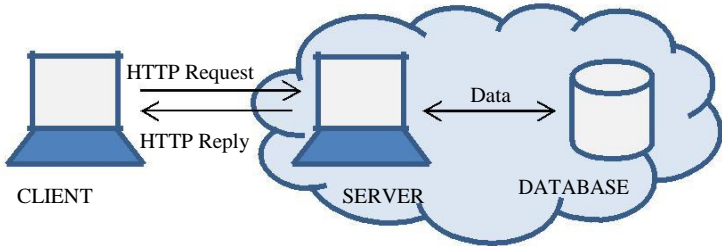


Figure 1 Web Application Architecture

Web Development involves programming both client and server side. Programming at server side is associated with writing web applications, providing web site, web pages, etc. Programming at client side is associated with developing tools or interfaces for accessing these web applications, sites or pages. Server side programming languages include Python, PHP, Ruby, ASP.NET, Java, CGI (C, Perl). Client side programming languages include HTML, CSS, and Javascript [6].

The rest of the paper is divided into three sections. Section 2 discusses web development using Python through CGI, WSGI, and Python frameworks. Section 3 compares Python with popular web programming languages, PHP and Ruby. Section 4 concludes the paper by giving summary.

2. WEB SERVER APPLICATION DEVELOPMENT

A. CGI and WSGI - Web Server Application Development using Python Standard Library

A.1 CGI

In a static web, documents (static web pages) over the web can be returned easily to the client on its request. However, dynamic web involves generating the reply on the fly based upon the user input (such as input in a form). Since, server has the ability to only take the user request and to return the response; it cannot handle user-specific data and generate the response. So, it uses web applications for dynamically generating response.

Creation of a web server needs a base server and a handler. Base server is responsible for carrying out HTTP communication between client and server, whereas handler is responsible for processing the request and returning the result in the form of a web page or document [7]. The module BaseHTTPServer has the common base server class

HTTPServer available in it. Within the same module, basic handler termed BASEHTTPRequestHandler is available but it offers only the functionality to take the client input. SimpleHTTPRequestHandler found in SimpleHTTPServer module offers additional GET functionality [7]. GET method is used for fetching the requisite document from the database. Parameters of GET method are part of URL. Similar to GET method is the POST method that is used for updating data. POST parameters are included in the body of the document [8]. CGIHTTPRequestHandler found in CGIHTTPServer module takes SimpleHTTPRequestHandler and provides additional functionality of POST method [7].

When a web server receives a client request via GET or POST method, it invokes the web application, and returns the dynamically generated HTML page to the client. This process takes place through CGI. CGI is the Common Gateway Interface, which act as an interface between server and the application program [7] [9] [10]. Figure 2 presents overview of working of CGI [7]. CGI application depicted in Figure may use database for data storage and retrieval purpose.

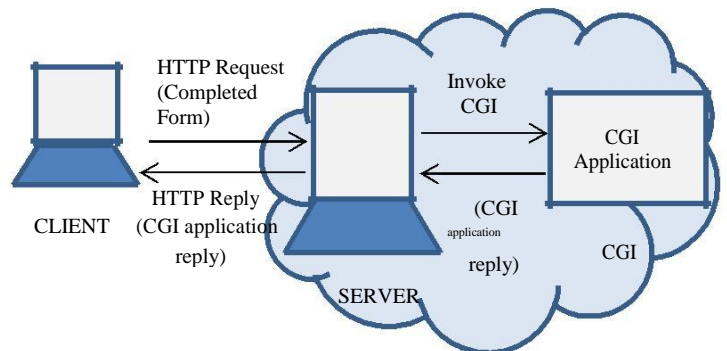


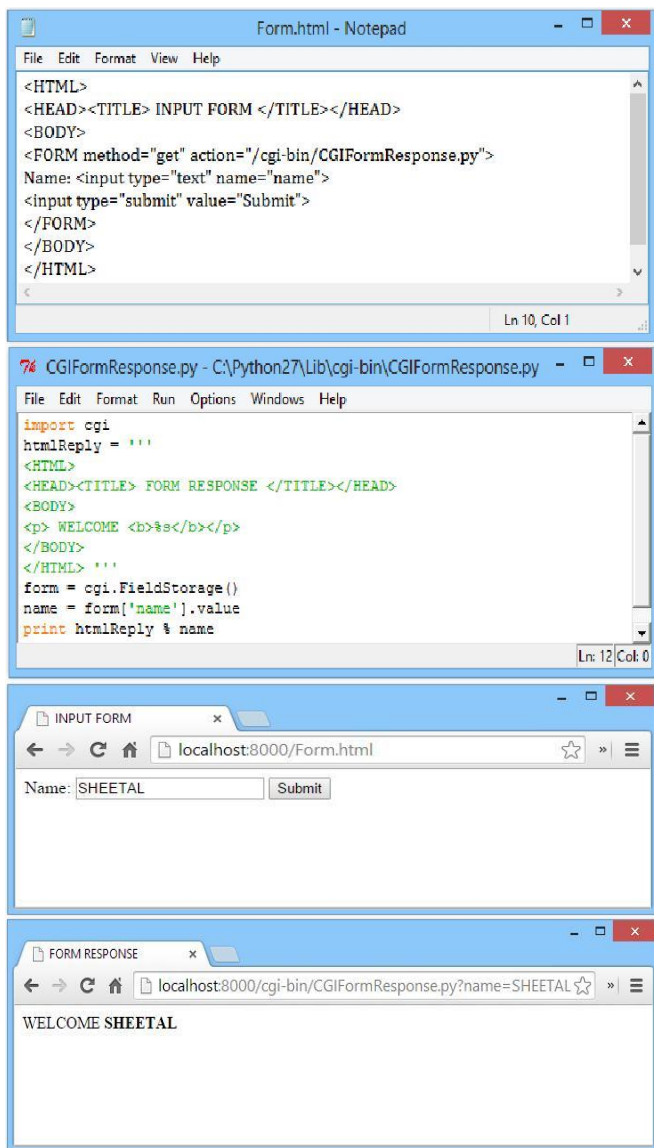
Figure 2 Working of CGI

Figure 3 depicts an HTML form (Form.html) that prompts the user for the input. Python CGI script (CGIFormResponse.py) is invoked after submitting the form. It should be noted that a server listening to user requests must be running all the time. For achieving this, you may run the server for Python2.x versions by executing CGIHTTPServer.py listed in Lib directory of Python2x folder from command prompt. For example,

```
C:\Python27\Lib>CGIHTTPServer.py
Serving HTTP on 0.0.0.0 port 8000 ...
```

It may be noted that above statement will run HTTP server on port number 8000. However, you may also run the CGIHTTPServer on a specific port. The most commonly used server is Apache. As specified, CGIHTTPServer module includes a class CGIHTTPRequestHandler extending handler class of SimpleHTTPServer module. This module handles the user request by forking a process and providing implementation for GET and POST method.

HTML form that is presented to user for providing input must be placed within the same folder where server is running (Lib). Within that folder, another folder is created with name cgi-bin, and .py files are placed there. It may be noted from Form.html file in Figure 3 that after submitting the form, CGIFormResponse.py is invoked for processing the input as also illustrated through URL in the browser. CGI python script imports cgi module which contains FieldStorage class [7] [11]. This class is needed for reading the data received from web client through web server. When python script is invoked, an instance of this class is created. This class instance comprise of key-value pairs that form a dictionary. Keys are the names of input fields on the form and values refer to the input provided by user for those fields.



Application based on CGI The WSGI application is called from the server side and two

CGI provides support for use of cookies using which server saves data on client side. It also supports processing of multiple inputs provided for a field, for example using checkboxes [7]. Web programming using CGI is not preferred since for each client request, server forks a process of CGI python program. This will lead to wastage of time since python interpreter will be initiated for each request. For a large number of requests, this may bring the server to halt. The problem associated with CGI can be overcome by using two modes– embedded or daemon [4]. Embedded mode eliminates the need to fork the process for every client request by integrating the interpreter within the server, for example – mod_wsgi and mod_python. In daemon mode, web server communicates with long running background processes termed daemons for processing request, for example – mod_wsgi and mod_fastcgi. This saves the time involved in spawning the process for each request.

A.2 WSGI

Web Server Gateway Interface (WSGI) is a standard that facilitate interaction between the server and the application by acting as an interface [4]. It is currently the most preferred interface for python web programming. Figure 4 presents overview of working of WSGI.

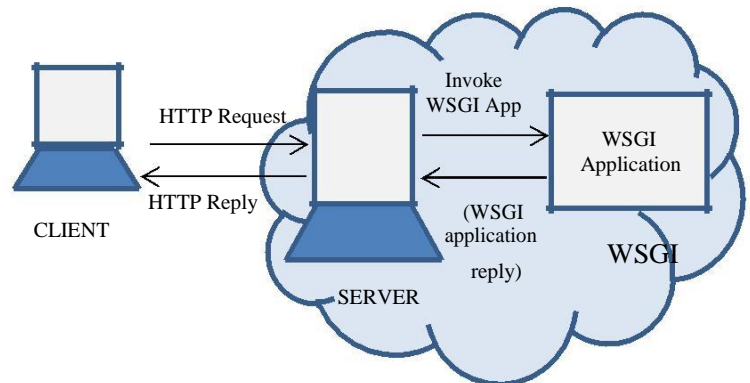


Figure 4 Working of WSGI

A WSGI application is an invocable application that takes two parameters [12]. First parameter is a dictionary type that comprises of environment variables such as HTTP_HOST, SERVER_PROTOCOL and their values [7]. Second parameter is a callable function that must be executed for initiating response to the client. The response prepared contains response code indicating status of response which comprises of status code and reason phrase (200 OK, 302 Found, 403 Forbidden, 500 Server Error). HTTP response may also contain HTTP headers such as Date (date and time of message initiation), Server (server software and its version), Content-Type (type of content in response body) and Content-Length (size of content in response body) [13].

Figure 3 Web required arguments are passed to it. Result returned by the

application is sent back to the client. It may be noticed that WSGI acts as an interface between the application and the server. The program at server side that invokes WSGI application can either be written by the developer or he may use the python provided reference server available in library `wsgiref.simple_server.WSGIServer` [7]. Thus, using WSGI one can implement both sides of the interface, server as well as application.

Figure 5 shows WSGI.py where method `make_server` of module `simple_server` creates a server listening on host `localhost` at port `8000`, responsible for calling function `wsgiApp`. This server is made to run in an infinite loop using method `serve_forever`. Function `wsgiApp` initiates HTTP response to the client by setting status to `'200 OK'` and header `'Content-type'` to `'text/html'`. Header `'Content-type'` helps the client in determining how to interpret the response of server. After initiating the response, function `wsgiApp` returns the content to the client.

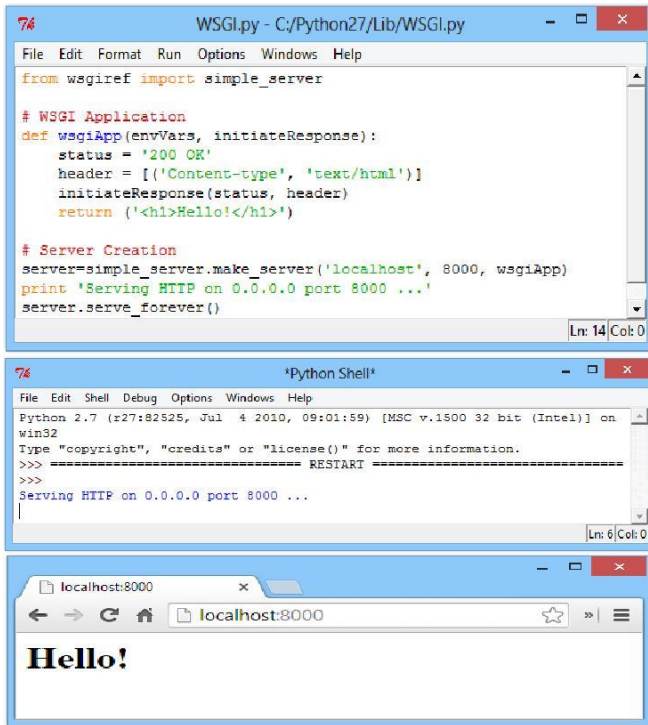


Figure 5 Web Application based on WSGI

Sometimes the request given to WSGI application may require some pre-processing such as modifying the request parameters or redirecting that request. Also, result returned by WSGI application may need further post-processing such as formatting the results. Pre-processing and post-processing involved with WSGI application can be achieved through additional wrapping functionality in form of layers termed as middleware [4] [14].

B. Web Server Application Development using Python Web Frameworks

Web programming using CGI and WSGI requires building web applications from the scratch by using Python standard libraries. Python provides with web frameworks [15] [4] in the form of packages/ modules that simplify the task of writing application programs. These frameworks lighten tedious job of developers. They support server and client side programming by providing support for several activities such as request interpretation (getting form parameters, handling cookies and sessions), response generation (generating data in HTML or other format such as pdf, excel), and storing data. The web frameworks are further categorized as full-stack and non-full-stack frameworks [15]. Full-stack frameworks provide components for every phase of programming in contrast to non-full-stack frameworks.

All the frameworks include templates and data persistence as key ingredients for constructing web [4]. Templates are used to avoid complex code that results when HTML and Python code is mixed in a single file. Templates are HTML files with placeholder for the data depending upon user input. Data persistence deals with storing and retrieving data and maintaining consistency. The data can be stored and maintained using plain text files, relational database engines such as MySQL, Oracle, or some object oriented databases. The web framework providing support for WSGI should be preferred [4]. This makes deploying an application easier.

In the following paragraphs, we briefly describe some popular web frameworks:

Full-Stack Frameworks

a. Django

Being a full stack framework, Django supports quick web application development requiring lesser core to be done. It is popularly known as “the web framework for perfectionists with deadlines” [16]. It provides ease in creating web applications with fewer lines of code and is scalable [17]. It includes a built-in server that can be used for developing and testing the applications. The framework comes with comprehensive, well-written documentation. Features of this framework include templates, support for relational database model (Databases include MySQL, SQLite, Oracle, and PostgreSQL), comprehensive security, etc. It is well-suited for database driven applications. The framework is based on the principle of reusability of code and non-redundancy of information. Examples of applications built using this framework include Pinterest, Instagram, Mozilla, The Onion.

b. TurboGears

TurboGears framework combines SQLAlchemy, Pylons, Genshi, Repoze, and Tosca Widgets. This framework includes 1.x and 2.x series where the 2.x series comprises of all the above specified components. However, 1.x series consists of SQLAlchemy, Genshi, CherryPy, and MonchKit. It supports both client and server side web programming. It offers designer friendly template system [18]. It has found its use mainly for offering solutions for industrial problems with high complexity. The framework supports several databases among which SQLAlchemy is considered to be most powerful database management system [19]. This framework is used by SourceForge, Fedora Community, TavolaClandestina, Glossom, ShowMeDo, etc.

c. web2py

The framework supports fast development of scalable, secure and portable web applications [20][21]. It does not require any installation and can be run via USB drive. This framework has no dependencies. The web2py application itself acts as an Integrated Development Environment offering all the capabilities such as creating application, debugging, and testing. It comes with a ticketing system that helps in tracking the source of error occurred by assigning a ticket to the user [22]. It supports several databases such as MySQL, PostgreSQL, SQLite, Oracle. Examples of applications built using this framework include Movuca, Instant Press, and LinkFindr.

Other full-stack frameworks include CubicWeb, Django Hotsauce, Giotto, Grok, Pylons, Wheezy Web, Gizmo, Glashammer, Karrigell, Kiss.py, Lino, Nigare, Porcupine, Pylatte, QP, SkunkWeb, Spyce, Tipfy, Tornado, Watson, Webapp2, WebBot, WebCore, Web.py, Webware, Werkzeug, and WHIFF [15].

Non-Full Stack Frameworks

a. Bottle

Bottle is an easy and simple framework used for quickly building small web-applications. This framework has no dependencies apart from Python Standard Library [23][24]. It has built-in web server that may be used for testing purpose. This framework is compatible with Python 3 [23]. Application built using this framework includes Pasttle.

b. CherryPy

CherryPy is an object-oriented HTTP web framework [25]. This framework was developed with the purpose of enabling user to develop the application as a regular python program with lesser lines of code. It is considered to be the one of minimal frameworks provided for Python [25]. It lacks templating system.

Built-in server of CherryPy offers a multi-threaded environment [27]. Websites build using this framework include Netflix, Urbanility, CherryMusic, Learnit Training [26].

c. Flask

Flask framework is based on Werkzeug, and Jinja 2, and good intensions [28]. This framework has no dependencies apart from Python Standard Library. Flask does not include components that need third party support such as validating form or providing a means of communication between the application and the database. However, such features may be added using extensions. Services offered by this framework include built-in HTTP server, support for unit testing, and RESTful web service [29]. Applications built using this framework are minitwit, flaskr, flask.pocoo.org etc.

d. Pyramid

Pyramid is a fast, reliable, simple, and mini web development framework [30]. This framework is compatible with Python 3. Being a fully documented framework, developers find an easy start for developing web applications. It supports several databases such as SQLAlchemy, Zope Object Database, and CouchDB. Websites built using this framework include newcars.com, SurveyMonkey, PwnedList.

Other non full stack frameworks include Albatross, Aquarium, AppWsgi, BlueBream, Bobo, circuits.web, Divmod Nevow, G-WAN, Gunstar, PyWebLib, Quixote, Spiked, Spinne, weblaver, WebStack, WSGIServlets [15].

Specifications of popular web frameworks are provided in Table 1 [31]. All the above mentioned web frameworks are based on MVC (Model View Controller) [32], architecture for web applications. It comprises of three components: model, view, and controller. Model comprises of the data and its logic i.e. requisite action to be performed. View refers to the interface presented to the user. Controller controls both model and view by calling a model depending upon user input and instructing model to update the data or change the state of data. Based on the result of invoked model, controller instructs the view to change the presented layout. For example, changing the view based on user input (such as page up and page down). Excluding Bottle and Flask, all frameworks mentioned above provide Object Relational Mapping (ORM) [33] that is responsible for mapping objects containing information to the relational databases where classes and objects are mapped to the relations and their tuples. So, ORM maps modifications and insertions for an object set to the relational database operations.

Framework	Latest Stable Version	License	MVC	ORM	Python
Django	1.6.5 (14 May 2014)	BSD	Yes	Yes	2.7 & above
TurboGears	2.3.2 (8 March 2014)	MIT	Yes	Yes	2.4 & above
web2py	2.9.5 (16 March 2014)	LGPL	Yes	Yes	2.4 & above
Bottle	0.12.7 (29 April 2014)	MIT	Yes	No	2.5 & above
CherryPy	3.3.0 (17 April 2014)	BSD	Yes	Yes	2.3 & above
Flask	0.10.1 (14 June 2013)	BSD	Yes	No	2.5 & above
Pyramid	1.4.6 (31 May 2014)	BSD	Yes	Yes	2.6 & above

Table 1 Comparison of Web Frameworks

Almost all the seven web frameworks specified above support features such as built-in server, internationalization (helps in adding languages and their translation), authentication and authorization of end-user, caching, debugging, error logging (help in keeping track of errors occurred along with their description, and time), routing the request to another URL, and managing sessions.

3. WEB DEVELOPMENT LANGUAGES COMPARED

There are several web development languages available in market that are used by the developers. These languages can be categorized as server side and client side scripting languages. Server side technologies include CGI, Java, ASP.NET, Ruby, Python, and PHP, whereas client side technologies include HTML, CSS, Javascript, and Java Applet. This section will discuss three most popular web development languages, Python, PHP, and Ruby, that are the preferred choices for a developer. All these languages are dynamic in nature and are open-source [34].

PHP [35] was developed by Rasmus Lerdorf in 1994. It stands for Hypertext Preprocessor. It is based on MVC architecture. There are several content management systems written using PHP that helps a non-programmer to build up an application. Examples include Joomla and Wordpress. PHP applications can easily be deployed. It offers great support for extensions and databases. Further, it is platform independent. It is not preferred if security is the prime concern. However, one may still incorporate security using

various appropriate measures [36] for reducing vulnerabilities. PHP based applications are not among the best at error handling. Also, PHP based applications are embedded in the server. It is used by big firms such as Facebook, Yahoo, Wikipedia, and Google.

Python was developed by Guido van Rossum in 1991. Python being a simple programming language, naturally lends itself to being a preferred choice for beginners. It offers fast development of web based applications that have comparatively lesser lines of code [9]. It is a modular and object oriented programming language. It is a flexible language because of the large availability of third party support. It follows strict indentation for defining the block of code. It used by big firms such as Google, pinterest, Youtube, and Instagram.

Ruby was developed by Yukihiro Matsumoto in 1995. It is also simple to learn, readable, maintainable, and a pure object-oriented language. One of the most important web frameworks of Ruby is Ruby on Rails which uses CGI as the gateway and has a built-in web server [37]. It can be used for scalable applications where security is one of the main concerns. It is used by big firms such as Github, Scribd, Twitter, and Groupon.

Table 2 compares these web programming languages on several criteria [38] [39] [40].

Comparison Criterion	Python	PHP	Ruby
Purpose	General	Web	General
Popular among developers according to TIOBE Feb 2014 index	8 th rank	7 th rank	14 th rank
Ease of Learning	Easiest	Easy	Difficult
development time	Faster	Fast	Fastest
Execution Speed	All are equally good		
Exception Handling	Best	Worse	Best
Security	Best	Worse	Best
Readable and Maintainable	Best	Worse	Worse
Robustness	Best	Worse	Better

Table 2 Comparison of Web Development Languages

4. CONCLUSION

Inspite of availability of so many web development languages, Python is evolving as the popular language among developers. Web development in Python can be done using CGI or WSGI that uses python standard libraries. However, such development requires building web applications from scratch. So, Python provides web development frameworks such as Django, Pyramid, web2py that helps in web application development. This helps the

developer in doing the minimal requisite core needed for building the applications.

REFERENCES

- [1] Berson, Alex. "Client-server architecture." No. IEEE-802. McGraw-Hill, 1992
- [2] Shklar, Leon, and Richard Rosen. "Web application architecture: Principles, Protocols and Practices." Editura Wiley (2009)
- [3] Suh, Woojong, ed. "Web engineering: principles and techniques." IGI Global, 2005.
- [4] Kubica, Marek. "How To Use Python in the web." <https://docs.python.org/2/howto/webrowsers.html>.
- [5] Grove, Ralph F. "Web Based Application Development." Jones & Bartlett Publishers, 2009
- [6] Pratt, Terrence W., Marvin V. Zelkowitz, and Tadepalli V. "Gopal. Programming languages: design and implementation." Englewood Cliffs: Prentice-Hall, 1984.
- [7] Chun, Wesley J. "Core Python Applications Programming." Prentice Hall Press, 2012.
- [8] Huffman, Steve and Bennett, Sean. "Web Development. How to Build a Blog". <https://www.udacity.com/course/cs253>.
- [9] Watters, Aaron, James C. Ahlstrom, and Guido Van Rossum. "Internet programming with Python." Henry Holt and Co., Inc., 1996.
- [10] Ramu, Chenna, and Christina Gemuend. "cgimodel: CGI programming made easy with Python." Linux journal 2000.75es (2000): 8.
- [11] "Web Python". http://webpython.codepoint.net/cgi_tutorial.
- [12] "PEP 333 – Python Web Server Gateway Interface v1.0." <https://www.python.org/dev/peps/pep-0333/>.
- [13] "Hypertext Transfer Protocol." <http://www.w3.org/Protocols/HTTP/1.0/spec.html>.
- [14] Gardner, James. "The Web Server Gateway Interface (WSGI)." The Definitive Guide to Pylons (2009): 369-388.
- [15] "Web Frameworks for Python." <https://wiki.python.org/moin/WebFrameworks>.
- [16] Forcier, Jeff, Paul Bissex, and Wesley Chun. "Python web development with Django." Addison-Wesley Professional, 2008.
- [17] "Django." <https://www.djangoproject.com/>.
- [18] "TurboGears." <http://turbogears.org/>.
- [19] Ramm, Mark, Kevin Dangoor, and Gigi Sayfan. "Rapid Web Applications with TurboGears." Pearson Education, 2006.
- [20] Mulone, Pablo Martin, and Mariano Reingart. "web2py Application Development Cookbook." Packt Publishing Ltd, 2012.
- [21] "web2py." <http://web2py.com/>.
- [22] Di Pierro, Massimo. "web2py Manual." 3rd ed. Lulu. com.
- [23] "Bottle:Python Web Framework." <http://bottlepy.org/docs/dev/index.html>.
- [24] "Full Stack Python. Bottle." <http://www.fullstackpython.com/bottle.html>.
- [25] "CherryPy." <http://cherrypy.org/>.
- [26] "Websites running atop CherryPy." <https://cherrypy.readthedocs.org/en/3.3.0/appendix/success.html>.
- [27] Hellegouarch, Sylvain. "CherryPy Essentials: Rapid Python Web Application Development." Packt Publishing Ltd, 2007.
- [28] "Flask." <http://flask.pocoo.org/>.
- [29] Grinberg, Miguel. Flask Web Development: Developing Web Applications with Python." O'Reilly Media, Inc.", 2014.
- [30] "Pyramid." <http://pyramid.readthedocs.org/>.
- [31] Index of Packages. <https://pypi.python.org/pypi/>.
- [32] Deacon, John. "Model-view-controller (mvc) architecture." <http://www.jdl.co.uk/briefings/MVC.pdf> (2009).
- [33] O'Neil, Elizabeth J. "Object/relational mapping 2008: hibernate and the entity data model (edm)." Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008.
- [34] Sebesta, Robert W. "Programming the world wide web." Pearson Addison Wesley, 2008.
- [35] Gerken, Till, and Tobias Ratschiller. "Web Application Development with PHP." New Riders Publishing, 2000.
- [36] Shiflett, Chris. Essential PHP security. "O'Reilly Media, Inc.", 2006.
- [37] Viswanathan, Viswa. "Rapid web application development: a Ruby on Rails tutorial." Software, IEEE 25.6 (2008): 98-106.
- [38] Purer, Klaus. "PHP vs. Python vs. Ruby–The web scripting language shootout." (2009).
- [39] Gaikwad H.M., Article on Code Wars: PHP vs Ruby vs Python languages
- [40] Hardy, Will. "Comparing web development platforms through the eyes of professional developers." Technical report, Institut für Informatik, Freie Universität Berlin, 2007.