

Blogging Web App – Using Django and Python

Mr. Alok Kumar Srivastava (Assistant Professor)

Janhvi Shukla, Arpit Tripathi, Kumar Prakhar, Krishna Mohan Pandey

Department of Computer Science & Engineering

Buddha Institute of Technology

GIDA, Gorakhpur, INDIA

Abstract: *The purpose of Blogging Web App is to automate the existing manual system by the help of computerized equipment and full-fledged computer hardware, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with. Blogging Web App will lead to error free, secure, reliable and fast management system. Thus, it will help the organization in better utilization of resources. Blogging Web App is a combination of both Blog as well as Novels. Blog contains the Information of various things related to Technology, Education, News, International, Business, Sports, and Entertainment. The main aim of this project is to provide data to students in only one site. Students can gather the information from one site as well as give their feedback and create their own blog. Students can post their views and thought and analyze themselves. Besides all such core functionalities, the application also includes features like FAQ, request, feedback etc. so as to provide a satisfactory user experience.*

Keywords: *Blogging, App, Data, Web-App, computerized, information.*

1. Introduction:

Blogging has become such a mania that a new blog is being created every second of every minute of every hour of every day. A blog is your best bet for a voice among the online crowd. Blogs were usually the work of a single individual occasionally of a small group, and often covered a single subject. Every citizen has the constitutional right to an education, which prepares them to play their role as a sophisticated member of society. With skilled and educated youths, education has the potential to change the course of a nation. A country must achieve a higher rate of economic growth in order to be classified as a developed country. A well-educated nation removes any impediments to growth and works hard to achieve long-term development. In the digital age, cutting-edge technology can be used to create a productive platform for academic purposes. Digitization has successfully revolutionized the learning process, which is impressive. Blogging is a relatively new educational tool. Blogging or weblogs are the evolution of traditional learning logs for students and teachers, whether as a supplement to traditional lectures or as an e-learning tool. The significance of these applications has grown as a result of changes in classroom dynamics that involve the substitution of traditional education for autonomous learning. In addition, the number of Open Universities and virtual environment courses offered by

traditional Higher Education Institutions (HEIs), which are potential users of weblogs, has increased in the last decade. More recently, "multi-author blogs" (MABs) have developed, with posts written by large numbers of authors and professionally edited. MABs from newspapers, other media outlets, universities, think tanks, advocacy groups, and similar institutions account for an increasing quantity of blog traffic. The rise of Twitter and other "microblogging" systems helps integrate MABs and single-author blogs into societal new streams. Blog can also be used as a verb, meaning to maintain or add content to a blog. A novel is a long, fictional narrative which describes intimate human experiences. BLOGGING WEB APP is a combination of both Blog as well as Novels. Blog contain the Information of various things related to Technology, Education, News, International, Business, Sports, and Entertainment The main aim of this project is to provide data to students in only one site. Students can gather the information from one site as well as give their feedback and create their own blog. Students can post their views and thought and analyze themselves. Besides all such core functionalities, the application also includes features like FAQ, request, feedback etc. so as to provide a satisfactory user experience.

The reason for Contributing to a blog Web Application is to mechanize the current manual framework by the assistance of automated gear and undeniable PC equipment, satisfying their necessities, so their significant information/data can be put away for a more drawn-out period with simple getting to and control of something similar. The necessary programming and equipment are effectively accessible and simple to work with. Publishing content to a blog Web Application will prompt mistake free, secure, solid and quick administration framework. Accordingly, it will help the association in better use of assets. Publishing content to a blog Web Application is a blend of both Blog just as Books. Blog contains the Data of different things identified with Innovation, Schooling, News, Worldwide, Business, Sports, and Amusement The principal point of this venture is to give information to understudies in just one website. Understudies can accumulate the data from one website just as give their criticism and make their own blog. Understudies can post their perspectives and thought and investigate themselves.

2. Literature Survey

Different work has effectively been done by different organizations as per writing for a blog or making a blog. These web sites give distinctive kind of offices yet does not have a couple. A few illustrations of writing for a blog sites are WordPress, Google's Blogger, in fact popular miniature contributing to a blog site Twitter and so forth Thus, considering forward the benefits and drawbacks of these pre-set up site a little exertion has been advanced to make an easy-to-understand site for contributing to a blog devotee. It's hard to give a definition for the possibility of a blog. Bruguera and Campàs suggest that it's anything but an individual or aggregate page wherein the adjusting and circulation system has been smoothed out to the point that the customer needn't mess with express data on the electronic medium or the modernized course of action to have the alternative to give content instantly, quickly and perpetually from any spot of Web affiliation (Bruguera and Campàs, 2007). Walker, on the other hand, portrays a blog as a webpage containing content favored with singular styles and presented in reverse dated consecutive solicitation (Herman, Jahn, and Ryan, 2010). Victor accepts a more specific perspective, portraying on the web diaries as a chain of significance of works, pictures, intelligent

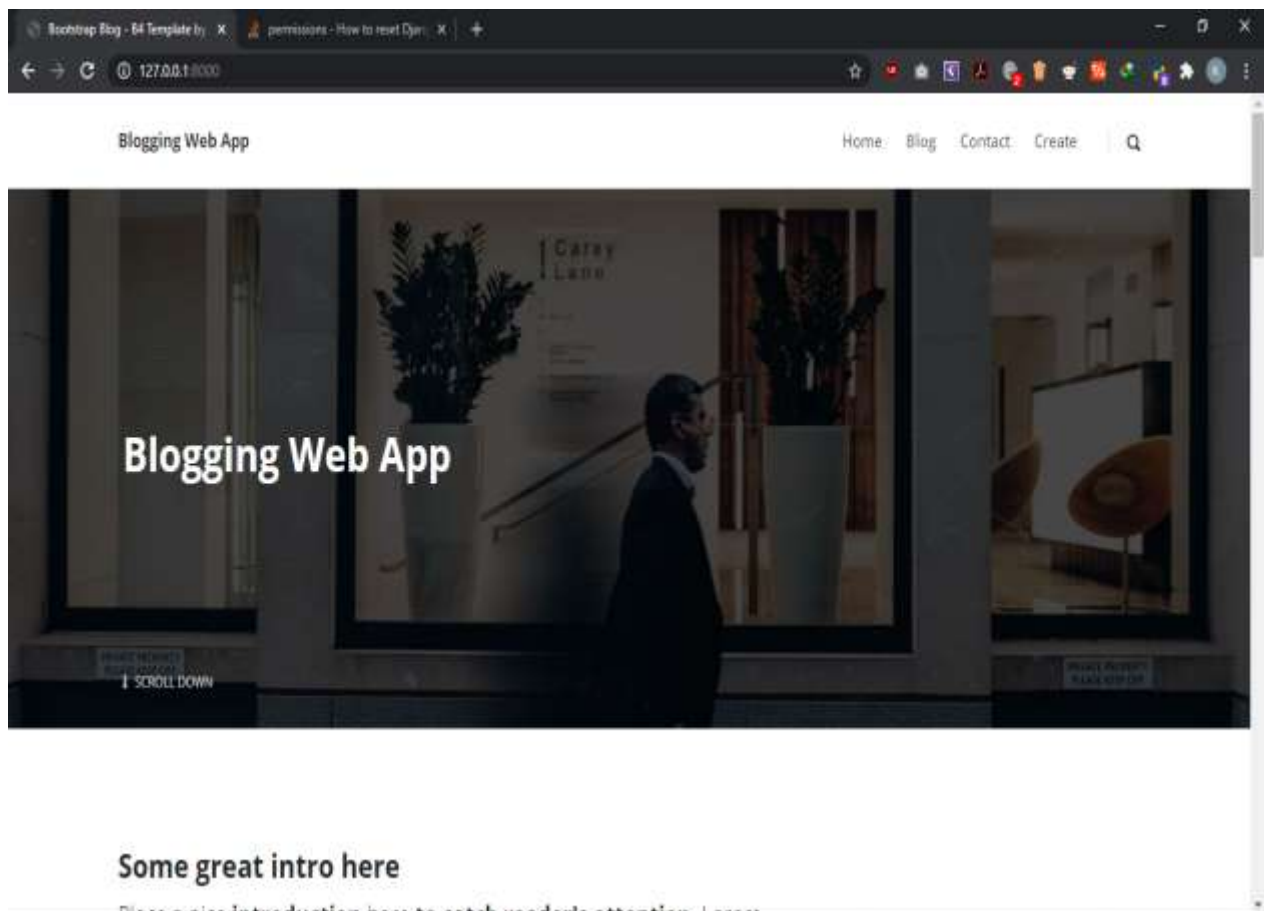
media objects and successively mentioned data that can be seen through a program (Winer, 2003). The interaction for making the blog is saved straightforward for the client. Preceding adding to a blog turned out to be notable, progressed networks took various designs including Usenet, business online organizations like Genie, Byte Information Exchange (BIX) and the early CompuServe, email records, and Notice Board Structures (BBS). During the 1990s, Web conversation programming made running conversations with "strings". Strings are powerful relationship between messages on a virtual "corkboard". From June 14, 1993, Mosaic Correspondences Venture cared for their "What's happening" overview of new destinations, revived step by step and archived month to month. The page was open by a phenomenal "What's happening" button in the Mosaic web program.

3. Django

Django's essential objective is to facilitate the production of complex, data set driven sites. The system stresses reusability and "pluggability" of parts, less code, low coupling, fast turn of events, and the rule of don't rehash the same thing. Python is utilized all through, in any event, for settings, records, and information models. Django additionally gives a discretionary authoritative make, read, refresh and erase interface that is produced powerfully through thoughtfulness and arranged by means of administrator models. Some well-known sites that use Django include

- Instagram
- Mozilla
- The Washington Post
- Disqus
- Bitbucket
- Mozilla Firefox
- Pinterest
- NASA
- Spotify
- YouTube
- Quora
- DropBox

Django was created in a high-speed newsroom environment; it was intended to make normal Web-developments undertakings quick and simple.

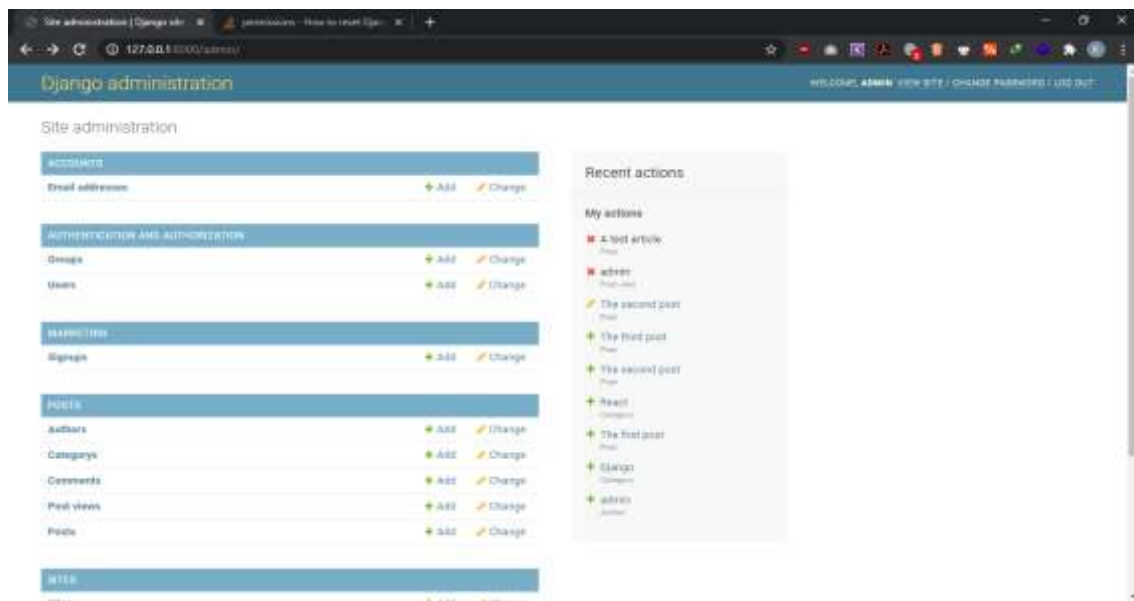


Django was made in the fall of 2003, when the web software engineers at the Lawrence Journal-World paper, Adrian Holovaty and Simon Willison, started utilizing Python to construct applications. Jacob Kaplan-Moss was recruited right off the bat in Django's improvement without further ado before Simon Willison's temporary position finished. It was delivered openly under a BSD permit in July 2005. The system was named after guitarist Django Reinhardt. Adrian Holovaty is a Romani jazz guitar player and a major devotee of Django Reinhardt.

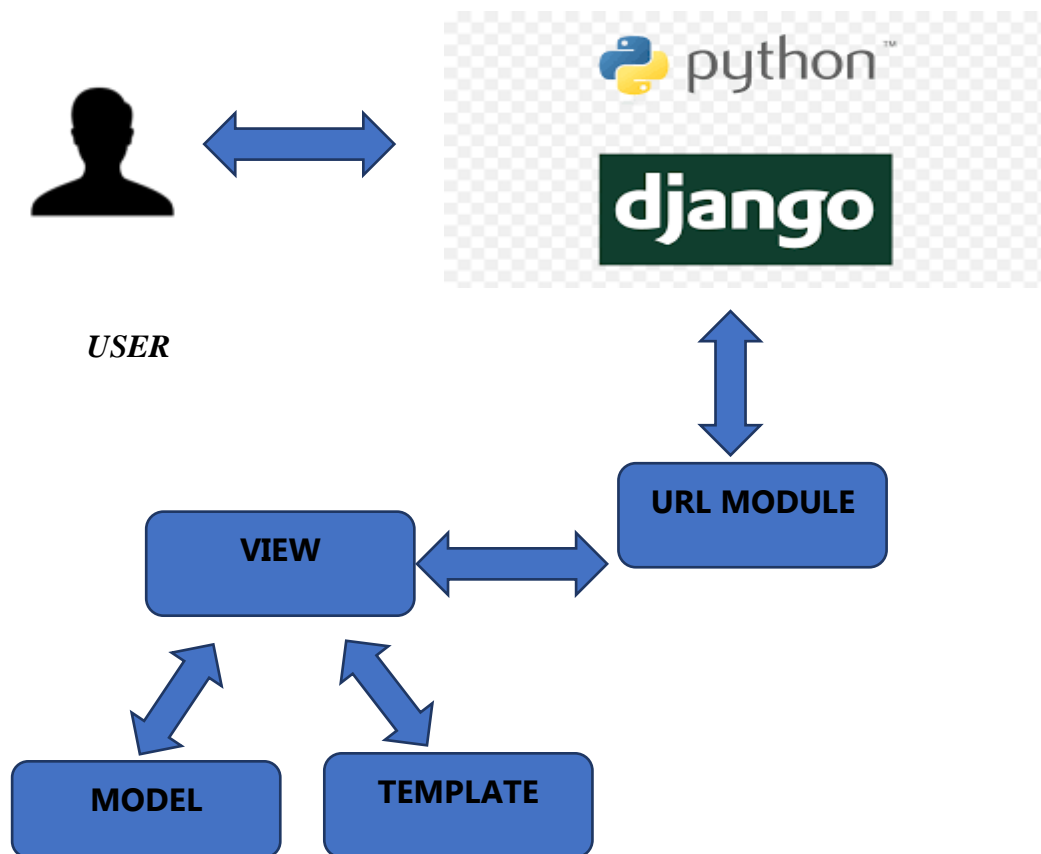
In June 2008, it was reported that a recently shaped Django Software Foundation (DSF) would keep up Django later on.

3.1. MVT Architecture

The MVT (Model View Template) is a product configuration design. It's anything but an assortment of three significant segments Model View and Template. The Model assists with taking care of information base. It's anything but an information access layer which handles the information.

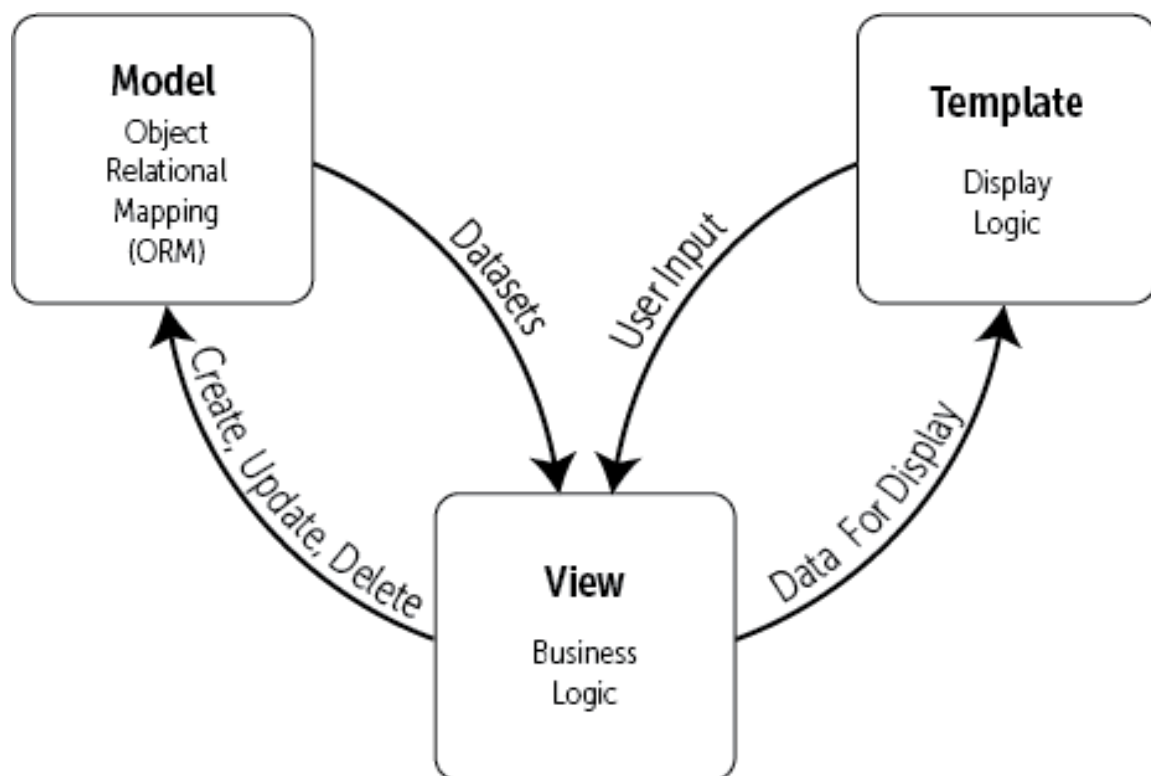


The Template is a show layer which handles UI part totally. The View is utilized to execute the business rationale and communicate with a model to convey information and renders a template. There is no different regulator and complete application depends on Model View and Template. That is the reason it is called MVT architecture.



In spite of the fact that Django follows MVC design yet keeps up its own shows. Thus, control is taken care of by the actual structure. Here, a client demands for an asset to the Django, Django functions as a regulator and check to the accessible asset in URL. On the off chance that URL maps, a view is called that collaborate with model and format, it's anything but a template. Django reacts back to the client and sends a template as a reaction. The developer gives the Model, the view and the template then, at that point simply maps it's anything but a URL and Django does the sorcery to serve it to the client. Here as shown in the above picture

1. The user sends a URL request for a resource to Django.
2. Django framework then searches for the URL resource.
3. If the URL path links up to a View, then that particular View is called.
4. The View will then interact with the Model and retrieve the appropriate data from the database.
5. The View then renders back an appropriate template along with the retrieved data to the user.

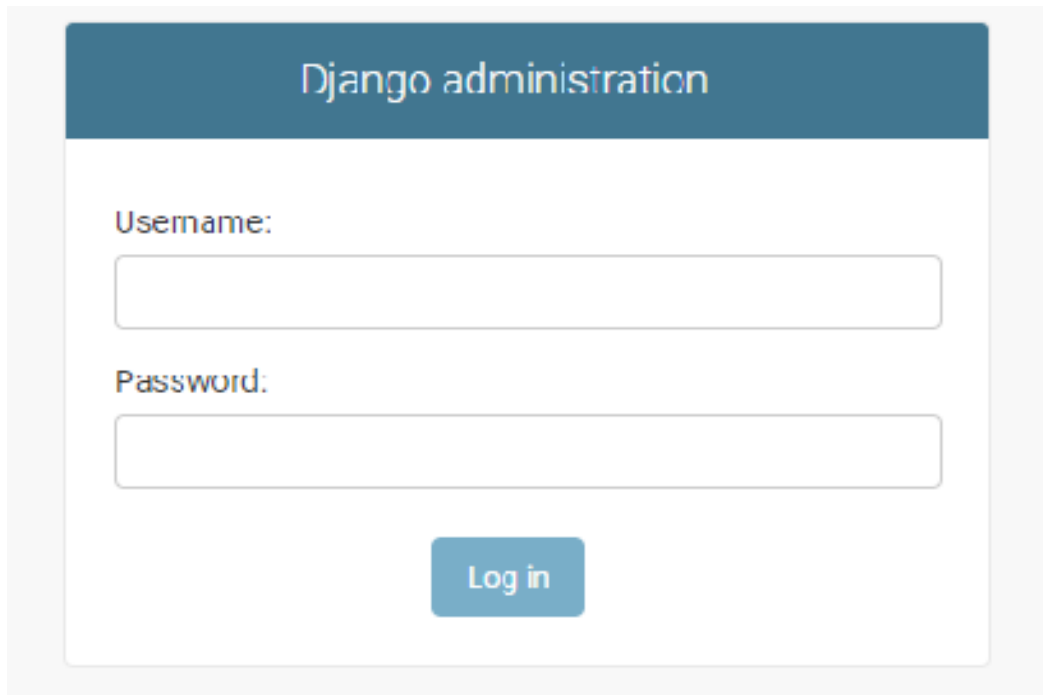


Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application. MVT Construction has the accompanying three sections –

- **Model:** Model will go about as the interface of your information. It is answerable for looking after information. It is the coherent information structure behind the whole application and is addressed by a data set (by and large social data sets like MySQL, Postgres).
- **View:** The View is the UI — what you find in your program when you render a site. It is addressed

by HTML/CSS/JavaScript.

- Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.



The image shows a screenshot of the Django administration login interface. At the top, there is a dark blue header with the text "Django administration" in white. Below the header, the page is white with a light gray border. There are two input fields: "Username:" followed by a text input box, and "Password:" followed by a text input box. Below these fields is a blue button with the text "Log in" in white.

4. Facilities required for proposed work

4.1 Software Requirements

Following are the minimum requirement for the system or the computer to carry out the proposed work. These are the minimum requirements for the proposed work to run smoothly, so here are the basic software requirements:

- Python 3.7 or higher.
- Django 2.1 or higher.
- SQLite 3 for Database.
- DB Browser for SQLite 3.2.0 or higher for viewing and modifying the database in a GUI manner (This is an optional software).
- Any Internet Browser (Chrome Recommended).
- An Operating System (OS), like Windows 7,8,10 and Linux OS like Ubuntu etc.

4.2 Hardware Requirements

Following are the minimum requirement for the system or the computer to carry out the proposed work as in terms of hardware. These are the minimum requirements for the proposed work to run smoothly, so here are the basic hardware requirements:

- Processor: 800MHz Intel Pentium 4 and equivalent or higher and AMD 6 gen. or above.
- Memory: 512 MB of RAM (minimum).
- Disk space: 250 MB of free disk space.

- Screen resolution is 1024x768 pixels. (For best user experience).

4.3 Recommended Hardware Configurations

Above hardware and software requirements are the minimum requirements for the proposed work, but for the best user requirement the user must have these hardware configurations so that this could give user the better experience:

- Processor: Intel Core i3 or equivalent or AMD 7 gen.
- Memory: 1 GB of RAM or higher.
- Disk space: 1.5 GB of free disk space.
- Screen resolution is 1024x768 pixels.

References

Journal Papers:

- [1]. Jenkins, Henry (2006). Fans, Bloggers, and Gamers. New York: New York University Press. p. 151. ISBN 978-0814742853.
- [2]. Bruns, Axel, and Joanne Jacobs, eds. Uses of Blogs, Peter Lang, New York, 2006. ISBN 0-8204-8124-6
- [3]. Rosenberg, Scott, Say Everything: how blogging Began, what it's becoming, and why it matters, New York : Crown Publishers, 2009. ISBN 978-0-307-45136-1
- [4]. Allen, Grant; Owens, Mike (November 5, 2010). The Definitive Guide to SQLite (2nd ed.). Apress. p. 368. ISBN 1-4302-3225-0
- [5]. "django/README". GitHub. Retrieved 8 September 2020.
- [6]. "FAQ: General - Django documentation - Django". Retrieved 30 April 2016.
- [7]. "django CMS - Enterprise Content Management with Django - django CMS". www.django-cms.org. Retrieved 11 November 2020.
- [8]. Osborn, Tracy (May 2015), Hello Web App (1st ed.), Tracy Osborn, p. 142, ISBN 0986365912
- [9]. Jaiswal, Sanjeev; Kumar, Ratan (22 June 2015), Learning Django Web Development (1st ed.), Packt, p. 405, ISBN 1783984406.