

Django: Python Web Framework - A Whitepaper



Introduction

Django is a high-level Python-based web framework that enforces fast evolution, clean design that embodies secure as well as maintainable websites. Django has been built by experienced developers; it takes care of a few daunting challenges faced during web development, so one can focus on writing his/her application without needing to regenerate the code. Django is an open source web framework that could help developer to efficiently compress overall web application development time and therefore costs. Python web developers mostly prefer Django whenever they build a highly customizable app, such as a social media websites and online shopping websites. When you need an app that can expand in depth with best support in any level of complexity and is capable of handling as many visitors and/or transactions as needed, Django is the solution. The important highlights of Django framework are reusability and “plug-ability” of components, less code, low coupling, and active development.

Aress software has a strong team of Django developers who have contributed in putting together this white paper.

Unique Features of Django

- **Python Web-framework:** Django is build using Python language. Python is the robust in nature with easy to learn and open source language. This feature makes Django a demanding choice for web development.
- **High Scalability:** Scalability refers to grow as per the need. The applications which are based on Django are Instagram, YouTube, Quora, Pinterest etc. The users of the applications have been growing day by day and the Django as a part of this application backend handle the data, process requests and conduct business logic operations.
- **Exceptionally Secure:** Django is secure because it covers the loopholes and help developer to avoid mistakes. It provides secure way to manage accounts.
- **Thoroughly Tested:** Django is the durable and powerful enough to withstand the dynamic changes happening in the industry. The community supports fully tested features with Django that makes the work bug free.
- **Fast Development:** We won't need expert backend knowledge to make a fully functional website using Django. Even no need of creating separate server files to design the database and connect the same while also making another file for transferring data to and from the server. Django itself handles this as well other plenty of tasks. We won't need extra files for each task.
- **Active Community Support:** Django helps developers to reduce web application development cost greatly. Also, it is supported by a large and active community of developers to resolve any issue raised while development.
- **Django Supports MVT Patterns:** This framework supports MVT pattern that allows small enterprises and businesses to speed up the development process of complex web applications by keeping their business logic and user interface layers isolated. Django enables programmers to handle templates while taking care of the controllers. Therefore, the developers are not required to focus on the interaction between the model and the view. They just have to match the pair of model, view, and template it to a specific URL.

Advantages

- **Object-Relational Mapping (ORM) support:** Django works like a bridge between the data model and the database engine. It supports an extensive set of database systems like Oracle, Postgres, MySQL etc. Django also supports NoSQL databases through Django-nonrel project (a project that making Django to run on Non-Relational Database System). For now, the only NoSQL databases supported are Google App Engine and MongoDB.
- **Multilingual Support:** The built-in internationalization system enables the Django to support multilingual websites development. Internationalization refers to the process of designing programs for the potential use of any locale or language and region. So, we can develop your website in the way that it will support huge set of languages as per the need.
- **In-built administration GUI and Development environment:** Django provides a well-structured ready-to-use user interface for administrative activities. Django also includes a lightweight web server to facilitate end-to-end application development and testing.
- **Model View Template architecture support:** Django is the Model View and Template (MVT) architecture-based web development framework. The Model act as a data access layer which handles and takes cares of the data. The Template act as a presentation layer which handles User Interface part thoroughly. The task of view is to run the business logic and interact with a model to import data and renders a template. Even though Django follows MVT pattern, it maintains its own conventions. The controller part is handled by framework itself.
- **Batteries included framework:** Django boast itself as a batteries-included framework. What that means is that it comes with a good amount of out of box features that you may or may not use depending on your application. Instead of having to write our own code, you just need to import the required packages that you want to use in your application development.

Django scores over competing technologies

Django Vs Laravel

Sr. No	Django	Laravel
1	Because of the uses of the Python programming language, Django has a fast performance than Laravel	Laravel uses the PHP programming language which is little bit slower as compared to Python.
2	Django supports the Model View Template (MVT) architecture. Controller is in-built in framework	Laravel supports the Model View Controller (MVC) architecture. Developer is required to write the controller code.
3	Django supports plenty of in-built middleware for the request-response processing like Gzip, Authentication, Security etc.	Laravel supports only HTTP middleware .
4	Django comes with ready-to-use administrative graphical user interface to handle administrative tasks.	Laravel does not come with any of the in-built administrative graphical user interface feature.
5	The Object Relational Mapper (ORM) support in the Django helps the developer to manipulate the database easily without knowing the database manipulation languages like SQL etc.	In case of Laravel, one must have to be aware about the database manipulation language for handling the bank-end database queries.
6	Django supports the 'Don't repeat Yourself'(DRY) mechanism , which reduces the repetition of work.	Laravel do not support 'Don't Repeat Yourself'(DRY) mechanism .

Django Vs Flask

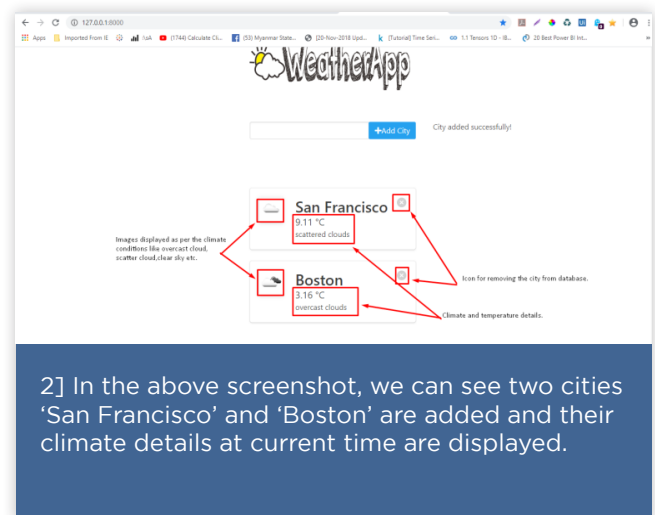
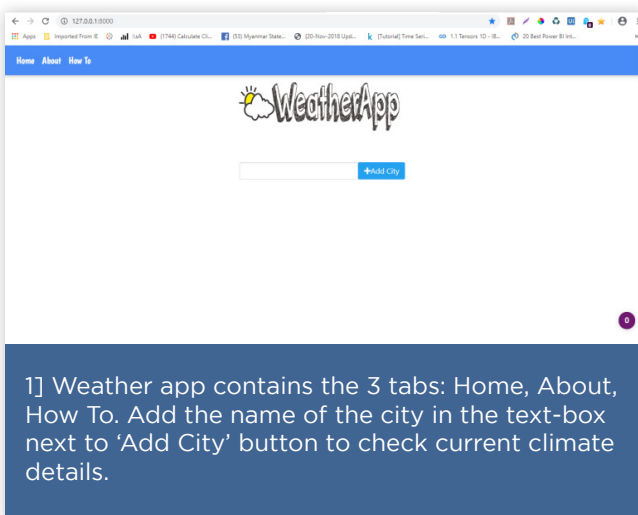
Sr. No	Django	Flask
1	Django is the full stack Python web development framework , where the plenty of things are included in framework.	Flask is the Python based micro web development framework , where whatever extensions need for project development you need to install it separately.
2	Since Django is the 'batteries-included' framework, it is most preferred for large project development .	Due to the lack of in-built features, flask is mostly preferred for small scale project development .
3	Due to larger community support for Django , it makes project development hassle-free.	Lesser community support . Hence the project development may face issues sometime.
4	Django is the good choice for building complicated sites with dynamically changing content such as Instagram etc.	Flask is best for building simple sites with the static content like blogs etc.
5	Django comes with below in-built features: 1] In-built admin panel /GUI feature, 2] Object relational mapper, 3] Middleware, 4] Database access layer	Flask does not support this feature. It needs to use SQL Alchemy or pure-SQL queries to perform the database related jobs.
6	Django is widely used for the REST-API development because of the plenty of functionality available in-built.	Since the lack of features and functionality, flask is not a good choice for REST-API development.

Illustration of Proof of Concept work done by Aress Software

Weather App:

Introduction:

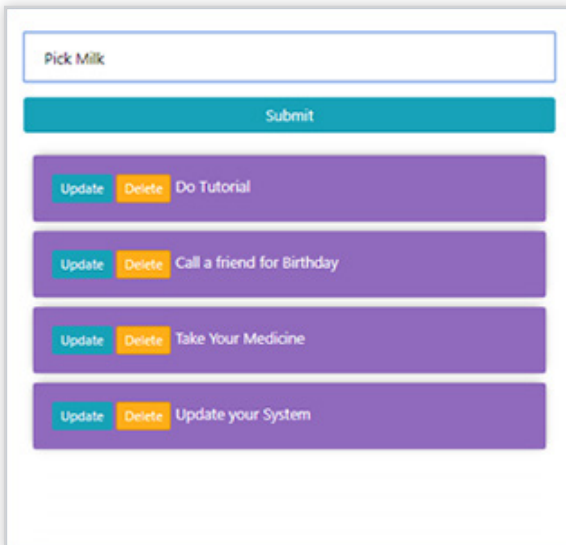
We have developed the simple app using Django and Python which would give accurate updates regarding the current weather situation in various cities around the world.



To-Do app:

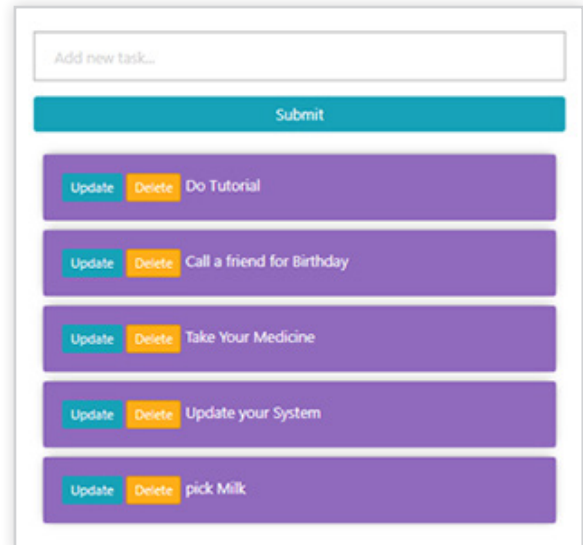
Introduction:

The To-Do app allows the user to maintain the list of daily tasks to do. As per the need user can make a list of his/her daily task by adding it, Update the task once completed, delete the wrong task or non-priority task from the list.



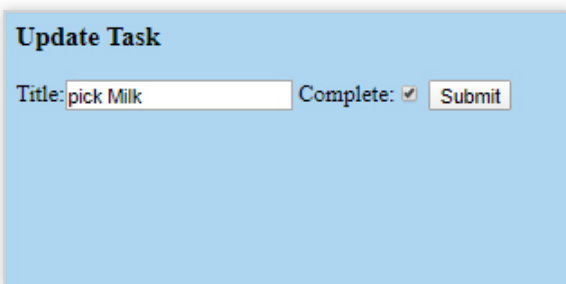
The screenshot shows a text input field containing 'Pick Milk'. Below it is a teal 'Submit' button. Underneath, there is a list of four tasks, each in a purple box with 'Update' and 'Delete' buttons: 'Do Tutorial', 'Call a friend for Birthday', 'Take Your Medicine', and 'Update your System'.

Add Task activity: Added the 'Pick Milk' task.



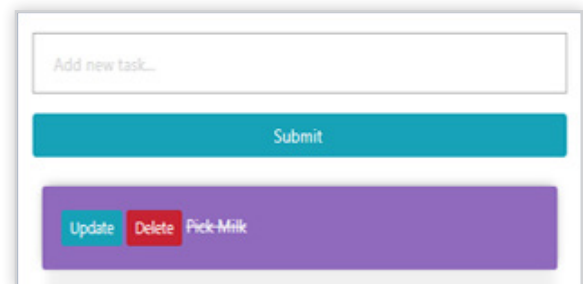
The screenshot shows the 'Add new task...' input field. Below it is a teal 'Submit' button. The list of tasks now includes 'pick Milk' at the bottom, in a purple box with 'Update' and 'Delete' buttons.

'Pick Milk' task added successfully at the end of the to-do list



The 'Update Task' form has a title 'Update Task'. It contains a 'Title:' label, an input field with 'pick Milk', a 'Complete:' checkbox which is checked, and a 'Submit' button.

Update task activity. Update the status complete & submit it.



The screenshot shows the 'Add new task...' input field and a teal 'Submit' button. The list of tasks now only contains 'Pick-Milk' in a purple box with 'Update' and 'Delete' buttons.

Delete task once it's complete. It will be strike-through and removed from to-do list.

Conclusion

Django is a high-level web framework which was created for quick and easy web application development. It supports transparent and high-quality code writing, making it important for developers as well as customers. There are number of advantages and a few disadvantages associated with Django. The prime objective for using Django should to take the developers from project conception to the launch platform quickly, and it really helps in 'Time to deliver' web application projects. It also takes away the burden of security issues like cross-site request forgery, SQL injection, cross-site scripting, and click-jacking and provides for strong security features. User authentication system enables developers to manage user accounts and passwords reliably. A key disadvantage is that at present it doesn't support real time web application development.

Aress software has a strong team of Django developers who could help in accelerating web application development to ensure scalability, extensibility along with strong security features.

Contact Us

✉ info@ares.ai