EasyChair Preprint

№ 4848

To-do Application using Python and Django

Mani Sindhuja Kalvakota, Gifty Suneeth and
Arun Kumar Reddy Chitimireddy

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

January 2, 2021

TO – DO APPLICATION

USING PYTHON AND DJANGO

ABSTRACT:

This TO-DO Application is a simple project developed using Django Framework, HTML and CSS. In this project user can add a task according to their routine and priorities. After completing their task the user can easily delete their task from the to do list so they can easily see what task is left for the user to do. Also, the user can give the title name according to task which makes it easy for the user to understand and tells the user what that task specifies at the end of day. Also, the user can easily mark which task is done and which is task is to done. Here user can mark undone task which remain to done so next day they can only see undone task. This application is to useful to user who has busy work schedules and lots of tasks in day.

KEYWORDS:

Django Framework, Model view Template (MVT), Model View Controller (MCV), SQLite Library, Django Template Language (DTL), Colour coding, Django's ORM, PyCharm.

INTRODUCTION:

- A to-do list is a simple list of tasks that allows you to visually see what you need to accomplish. It can be very useful for managing time, by planning your day ahead of time, and prioritizing activities.

- It's all in one place for easy reference! Grows and shortens as you work through items. Posted in a common place so you don't forget your tasks. It is an organizational scheduling tool.

TECHNICAL SPECIFICATIONS:

Technologies Used: Various front-end and back-end technologies are available in this era of digitalization. Our Project is done using Python, Django framework and a little bit of HTML and CSS.

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before

executing it. This is similar to PERL and PHP.

- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.
- Python's features include:
- Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read − Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain − Python's source code is fairly easy-to-maintain.
- A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases − Python provides interfaces to all major commercial databases.
- GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable − Python provides a better structure and support for large programs than shell scripting.
- Apart from the above-mentioned features, Python has a big list of good features, few are listed below −
- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Django Framework:

- Django runs on an MVT (Model View Template) system which is different from MVC (Model View Controller) that powers Laravel. Django is a Python web framework that simplifies common practices in web development.
- Model: The model in Django sets out the schema for our database. With Django's ORM, you can declare the fields, field types and any extra data for instance meta information.
- View: The View in Django, is where you set all of your code logic and

algorithms. With the View, you can get some results from the database or manipulate some data. The view basically expects a request and a response. The response is usually a HTTP redirect, HTTP error (404), Mime Types or a call to load a template.

- Template: The template in Django is the plain HTML code with Django's Template Language in it. DTL (Django Template Language) is the language you'll be using to dynamically write your template. You can do almost everything you'd with python and any other language.

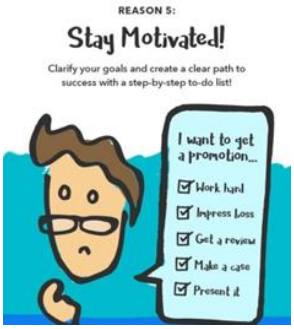
Front-End Technologies:

HTML: HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

CSS: CSS (Powell, 2010) stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on the screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once.

SQLITE3: SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

EXISTING METHODS:



- Reorder Tasks: This one is really simple. One of the advantages of an app-based to do list is that you are not limited by ink on paper. Yet, many apps won't let you easily re-organize the order of the to do's on your list.
- Prioritization: This is another basic time management feature. It amazes how many apps don't allow prioritization of tasks. (Or have it poorly implemented.) It is important to be able to separate important tasks from the lesser ones. This should be by default.
- Quick Entry/Capture: To do apps should always be ready to quickly capture a random task. It shouldn't require 16 steps to enter a to do. They should be quick to allow you to add a to do and get it "out of your head."
- Simple UI: Following on from quick capture, overall the app should be simple. Simple to use. Simple user interface. Too many features just clutter things up. And 99% of those extra bells and whistles never get used.
- Future Tasks: To do lists should remember so you don't have to. Tasks should be able to "send tasks to a future date" and then be out-of-mind

until that date. Some of the bigger apps have this, but I would like to see it be easier to use and more intuitive.

- Multiple Tasks: Multiple tasks can be edited and also gives multiple alerts at the same time.
- Drag and Drop: It moves your task from one list or project to another with ease.
- Filters and Sorting: Select by filter and then arrange tasks by due date, alphabetically, by tag or by context.
- Starred tasks: Can give some of your to-dos VIP status by adding a star to the most important tasks to have them quickly jump to top of the list.

PROPOSED SYSTEMS:

- Title name: Adding title name to the task gives easy glance of the content.
- Colour coding: Adding colours to each task list to make it easier to identify which lists you are viewing, for example blue for work, brown for home, etc.
- Marking: To mark a task that is completed with tick and also, it marks a task that is incomplete with the cross “X”.
- Calendar view: Get a better perspective of how busy your schedule is withdrawing view. Tap and select the filter the tasks shown or drag and drop tasks to the calendar to assign them due date.
- List arrangement: A list can be ordered, reordered accordingly.
- List icons: Can choose a different icon for each of your lists.
- Setting alarm: An alarm can be set so that one cannot forget and reminds you at the right time without delay.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System : intel i5
- Hard disk : 1 TB
- Monitor : 15’’led
- Input devices : keyboard, mouse
- Ram : 8GB

SOFTWARE REQUIREMENTS:

- Operating system : windows 10 Coding language : Python
- Tool : PyCharm, Visual studio code
- Database : sqlite3

REFERENCES:

https://www.digitalocean.com/community/tutorials/build-a-to-do-application-using-django-and-react

https://www.pythonistaplanet.com/to-do-list-app-using-django/

https://www.skyfilabs.com/project-ideas/todo-web-app-using-django

https://github.com/shacker/django-todo

https://www.geeksforgeeks.org/python-todo-webapp-using-django/

https://zapier.com/blog/best-todo-list-apps/

https://appigo.com/todo-cloud/

AUTHORS:

1. C. Arun Kumar Reddy is currently pursuing his B. Tech specialised in “Computer Science and Engineering” at Malla Reddy College of Engineering, JNTU, Hyderabad.
   He is an active team player and has developed an application “To – Do” using Python and Django framework. His interests include learning about the evolving technologies, Machine Learning, Data Science and Artificial Intelligence.
2. K. Mani Sindhuja is pursuing her B. Tech in “Computer Science and Engineering” at Malla Reddy College of Engineering, JNTU, Hyderabad.
   She has developed a mini project which is a “To – do applocation” developed using Python and Django framework. Her interests include artificial intelligence and internet of things.
3. Gifty Suneeth is pursuing her B. Tech in “Computer Science and Engineering” at Malla Reddy College of Engineering, JNTU, Hyderabad.
   She has developed a mini project which is a “To – do applocation” developed using Python and Django framework. Her interests include artificial intelligence, data science and internet of things.