Project Report
ON
DLock - Document Digital Locker

Submitted in the partial fulfillment of the requirement for the awards of the degree of the Master of Computer Application (MCA)

Session: 2018-2020

UTD, Bhopal

Submitted by:

Annanya Sharma
Enroll: AT9910999013
MCA VIth sem

Project Supervisor:

Prof. Dr. C.P. Agarwal
(H.O.D)

Submitted to:
DEPARTMENT OF COMPUTER APPLICATIONS

Makhanlal Chaturvedi National University of Journalism and Communication Bhopal

SELF CERTIFICATE

This is to certify that the Major Project report entitled "**DLock - Document Digital Locker**" submitted to **Makhanlal Chaturvedi National University of Journalism and Communication,** in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications (MCA), is original work carried out by myself **Ms. Annanya Sharma** with **Enrollment no. AT9910999013** under the Supervision of **Prof. Dr. C.P. Agarwal**.

The matter embodied in this project is a genuine work done by myself and has not been submitted whether to this university or to any other university for the fulfillment of the requirement of any other course of study.

Date:

Signature of the Student

Annanya Sharma

Email: annanyasharma868@gmail.com

Mobile No.: 9584159300

Address: Mandideep, Near Bhopal, M.P

Signature of Supervisor

Prof. Dr. C.P. Agarwal

(H.O.D Computer Department)

DECLARATION

I hereby declare that this project report titled **"DLock - Document Digital Locker"** submitted to the Department of Computer Applications of the "**Makhanlal Chaturvedi National University of Journalism and Communication"** in partial fulfillment for the award of Master of Computer Applications(MCA) in Computer Science is based solely on my original work except for citations and quotations which have been duly acknowledged.

Annanya Sharma
MCA(LE)

ACKNOWLEDGEMENT

I would like to acknowledge and feel great pleasure to express my deepest rectitude and respect towards my internal project supervisor **Prof. Dr. C.P. Agarwal**(H.O.D Computer Department) of **Computer Science and Applications**, for his valuable support and guidance. He guided me throughout the process from conception and till the completion of this project.

I would also like to express my gratitude to respect my faculties **Dr. Manish Maheshwari, Dr. Anurag Seetha, Mr. Ravi Mohan Sharma, Dr. Sunita Dwivedi, Dr. Manoj Pachariya** for time to time valuable suggestion. I am Indebted to my external guide **Mr. Arvind Mishra Head** at **"Techlene Software Solutions Pvt. Ltd.".** Whose stimulating interest and preserving effort inspired me to complete my project. This Major Project is the result of contribution of many minds. I am very thankful for the open handed support extended by many people. While no list would be complete, it is my pleasure to acknowledge the assistance of my friends who provided encouragement, knowledge and constructive suggestions. Finally, I would like to thanks The **God** and my **Parents and Brother** whose blessings have always been with me and given me the strength to complete this project.

Annanya Sharma

MCA (LE) 2018 - 2020

Enroll No – AT9910999013

ABSTRACT

“DLock is a document digital locker which will provide secure access to all paper documents to easy access form everywhere in the world ”. It is aimed at eliminating the use of physical documents and enables sharing of verified electronic documents across Government agencies and college purpose. DLock will reduce the administrative overhead of college administration and Government agencies created due to paper work. It will also make it easy for the students to receive services by saving time and effort as their documents will now be available anytime, anywhere and can be shared electronically. Keywords: Databases, Programming I.

LANGUAGE AND TOOLS TO BE USED

- Processor:CORE i3,i5
- Hard Disk:126 GB
- RAM :8 GB D. Software Requirements
- Operating system: Windows XP/8/10
- Database : Sqlite
- Database connectivity: Django
- Scripting : Python,, bootstrap , Django
- Server Side:Python Django
- Python version : Python 3.7.8

INDEX

1. INTRODUCTION

DLock the idea of paperless documentation, DLock is a platform for issuance and verification of documents & certificates in a digital way, thus eliminating the use of physical documents. College members who sign up for a DLock account get a dedicated storage/space. Organizations that are registered with DLock can push electronic copies of documents and certificates (e.g. driving license, Voter ID, School certificates) directly into members lockers. Members can also upload scanned copies of their legacy documents in their accounts. These legacy documents can be electronically stored in our webapp.

1.1 PROJECT OBJECTIVE

Objectives of DLock:

- Enable digital empowerment of residents by providing them with DLock webapp.
- Enable e-Signing of documents and make them available electronically and online Minimize the use of physical documents
- Ensure authenticity of the e-documents and thereby eliminate usage of fake documents.
- Secure access to Govt. issued documents through a web portal for residents.
- Anytime, anywhere access to the documents by the resident Open and interoperable standards based architecture to support a well-structured standard document format to support easy sharing of documents across departments and agencies.
- Ensure privacy and authorized access to resident's data.

1.2 FUNCTIONALITIES

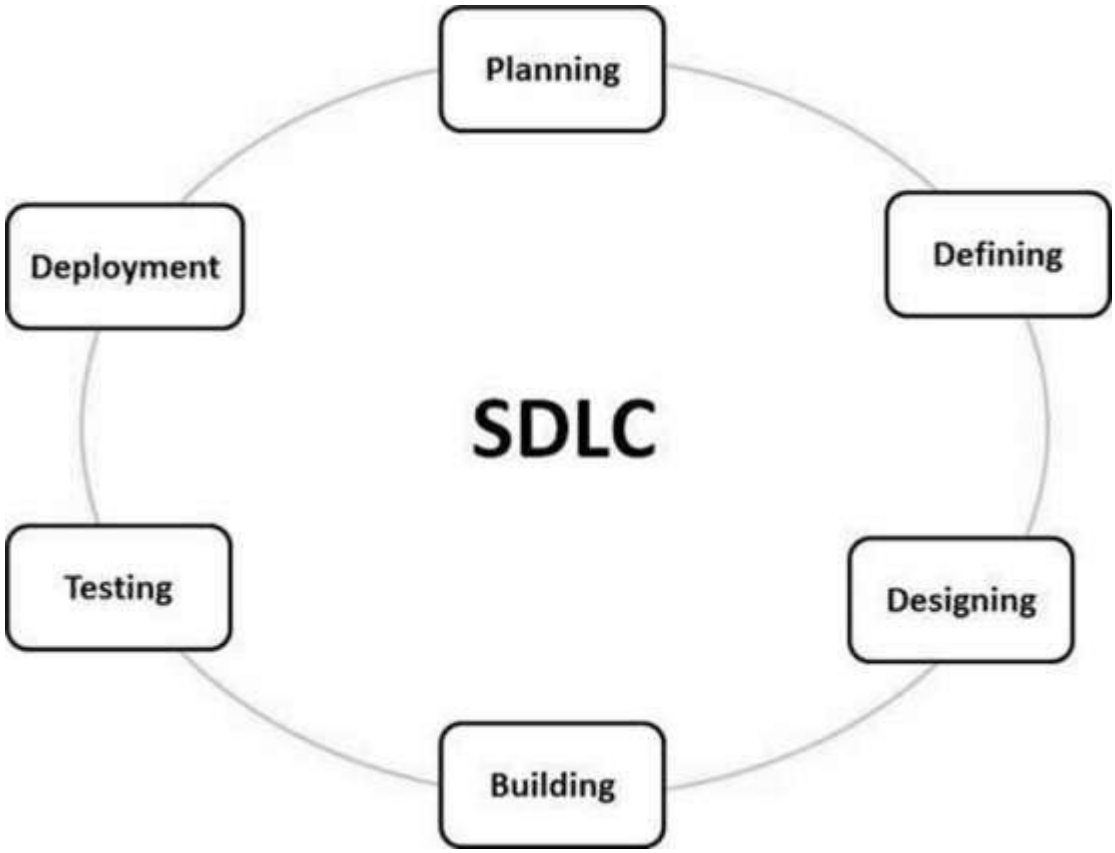
- To increase efficiency of managing the Govt. issued documents.
- It tracks all the information of a client.
- Manage information of client.
- Keeps record of the document

2. SOFTWARE DEVELOPMENT LIFE CYCLE

2.1 INTRODUCTION

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages −

Stage 2. 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2.2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 2.3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 2.4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 2.5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 2.6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT - User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

3. SYSTEM REQUIREMENT SPECIFICATION (SRS)

3.1 INTRODUCTION

DLock is a service to provide a secure dedicated personal electronic space for storing the documents of students. The space can be utilized for storing personal documents like University certificates, PAN cards, Voter ID cards, Hall tickets, College ID card. The service is intended to minimize the use of physical documents and to provide authenticity of the documents. It will also provide secure access to University and Government issued documents. It also makes it easy for students to receive services. It reduces the time spent by the students standing in queue to collect hardcopy of the documents from the administrative office. If there are any mistakes in the certificates, students can directly interact with admin through this application. Whenever the students want to share their documents with others, they can share them within a fraction of seconds.

3.2 Modules

The main users in the project can be categorized into three modules as follow:

3.2.1 User
3.2.2. Admin

RELATED WORK

(A). Purpose of the document this paper is the Software Requirement Specification (SRS) for the DLock system. The purpose of this paper is to describe the functionality, requirements and general interface of our project.

(B). Scope for development of this paper Students can show their hall tickets through this application if they loose it and also they can show their id proofs when required.

- Main Modules of the system.

3.1.1 USER

- In this module user has to register, then they can login and upload the user private documents such as Voter id, Aadhaar card, Passport etc., to their account according to the category.
- User can share their documents with admin and also with other students
- User can view and download the documents issued by the admin. When it is required.

- If any student lose their Certificates and if it is found by any other students, they can notify about the Certificate to all other students as well as to the admin when that has been handed over to the admin so that the student who have lost that particular certificate can come and collect it from admin.

3.1.2 Admin

- In this module admin will login, then he/she can add categories such as Marks card, Voter id, Hall ticket, College id card, etc.,
- Admin can send documents such as Marks card, Hall tickets, Study certificate etc., to students by selecting category and username.
- Admin can retrieve the documents uploaded by any student according to the category by selecting their username.
- Admin can send documents such as Study Certificate, Provisional marks card,Transfer Certificate etc., which are requested by the students in a issue Certificate form.
- Admin can view a notification sent by the student who found the lost document of another student and he will return it back to the student who lost it.

I. EXISTING SYSTEM

Current systems are manually collecting documents from administrative office and carry them when required. Limitaions of the existing system

There may be a chance of losing documents while carrying manually.

- Document will not be available during emergency(Anytime and Anywhere).
- Takes more time to receive documents directly from college administrative department.

II. PROPOSED SYSTEM

Is a web portal or mobile application where students can receive documents to their respective accounts directly from college administrative office through online.

Benefits of Proposed System

- Prevents loss of documents.
- Documents can be accessed anytime and anywhere.
- No need to wait till the hard copy of documents is received from administrative officer.

- Notifies lost document. International Journal For Technological Research In engineering Volume 3, Issue 9, May-2016 ISSN (Online): 2347 - 4718 www.ijtre.com Copyright 2016.All rights reserved. 2001
- Study certificate and Provisional marks card can also be accessed.

II. SPECIFIACTIONS

(A) Functional Requirements

1. Admin should be able to login to the system through the first page of the application
   - Can add the category.
   - Based on the Category Admin can send documents to the specific student.
   - View records which are uploaded by the students.
   - Send the documents like Study Certificates ,Tranfer Certificate, etc., that are requested by the students..

2. Students should be able to
   - Register and login to the system through the first page of the application
   - Upload their private documents according to the category.
   - Share documents with admin as well as with students.
   - View and download documents issued by the admin.
   - Can request for other documents like study certificate ,transfer certificate, etc., by filling Certificate form.
   - Can notify about lost document. B. Non-Functional Requirements

Reliability: The capability to maintain the specified level of performance is called reliability. Unauthorized person will not be able to access the details.

IV. Availability:

This system must be readily available to students who need to access their documents anytime.(24*7) The system must work relatively fast and must provide the data on request as soon as possible without affecting the quality and accuracy.

- Security:

This system must be highly secured and must authenticate users strictly. This system would require handling confidential data. This system must keep the documents with more security as they will be stored in student's private account.

- Maintainability:

Maintenance is typically done after the software development has been completed. As the time evolves, so do the requirements and needs. It revolves around the understanding of the existing software and the effects of the change.

- Portability:

Portability is the ability of the system or application that can run in various environments. As the web application is based on the java language, the application is portable.

V. SYSTEM DESIGN USECASE DIAGRAM

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication associations between the actors and users and generalization among use cases. ADMIN: Fig 3.1 Usecase Diagram for Admin USER: Fig 3.2 Usecase Diagram for User International Journal For Technological Research In Engineering Volume 3, Issue 9, May-2016 ISSN (Online): 2347 - 4718 www.ijtre.com Copyright 2016.All rights reserved. 2002

ADMIN:

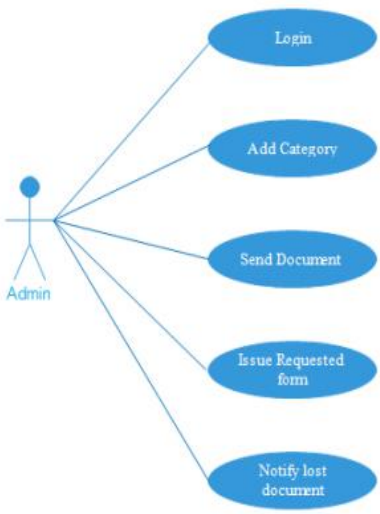


Fig 3.1 Usecase Diagram for Admin

STUDENT:

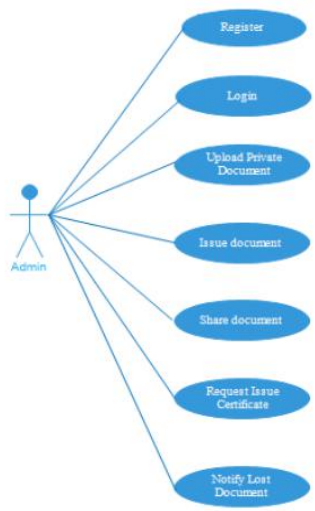


Fig 3.2 Usecase Diagram for Student

VII. CONCLUSION

This project can be used by the students to store their private documents safely with more security. Its main aim is to eliminate carrying of documents physically. Each student will have their own username and password to login to their respective accounts. Since admin issues documents directly to the students according to their USN, it overcomes fraudness and also it reduces the time spent by the students to collect hardcopy of the documents from the administrative office. It also makes it easy for students to receive services.

4. PROJECT PLAN

4.1 Scope management

- DLock will provide secure access to documents. It uses authenticity services provided by Userid's.
- DLock is aimed at eliminating the use of physical documents and enables sharing of verified electronic documents across government agencies.
- DLock provides a dedicated personal e-storage space to users.
- DLockwill reduce the administrative overheads of government departments and agencies created due to paper work. It will also make it easy for Indian users to receive services by saving time and effort as their documents will now be available anytime, anywhere and can be shared electronically.
- To sign up for your DLock, you need your Login details which you fill when you register in DLock.

4.2 Risk Management Plan

We at DLock providing full security of the data that we keep and security is our priority

4.2.1Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The increasing visibility of software as a system element and the attendant "costs" associated with a software failure are motivating forces for well planned through testing.

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery to customer. The goal is to design a series of test cases that have a high likelihood of finding errors but how? That' where software testing techniques enter the pictures.

4.2.2Testing Objectives

E Testing is a process of executing a program with the intent of finding an error. E A good test case is one that has a high probability of finding an as-yet-undiscovered error. E A successful test is that uncovers an as-yet-undiscovered error.

4.2.3 Testing Principle

All tests should be traceable to customer requirement. E Tests should be planned long before testing begins. E The Pareto principle applies to software testing. E Exhaustive testing is not possible. E To be most effective, an independent third party should conduct testing.

"Software testing involves executing an implementation of the software with test data and examining the outputs of the software and its operational behavior to check that it is performing as required. Testing is a dynamic technique of verification and validation because it works with an executable representation of the system"

4.2.5 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. Using the component – level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. The unit test is white-box oriented and the step can be conducted in parallel for multiple components.

E **Login Module:**

Sr. No.
Test Case Description **Input Expected**
Behavior **Observed behavior Test**
Result

1 Can ID field be
Null? Null ID ID cannot be
NULL
Warning msg "ID can't be NULL" Success

2 Can password be
Null? Null password Password Can't
be NULL
Warning msg "password can't be Null"
Success

3 Login button is
working or not?
Button pressed
Perform login processing
Call proxy Inbox frame Success

4.

Is Login Frame displaying properly?
Invoke Login Frame
All text fields are displayed and are properly aligned
Little alignment problem Success

4.2.6 Integration Testing

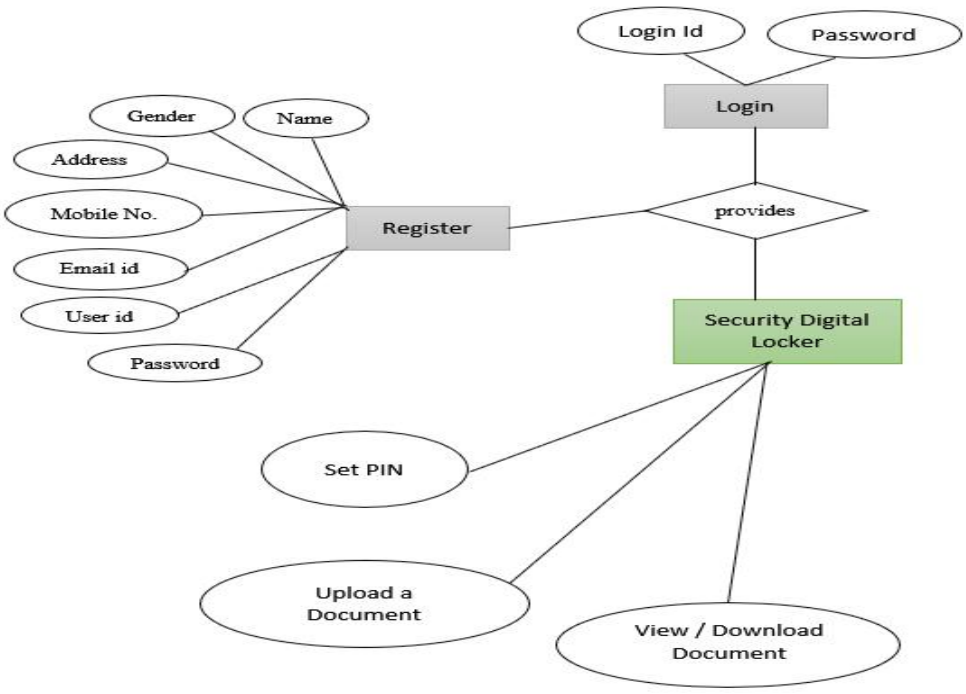
Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design.

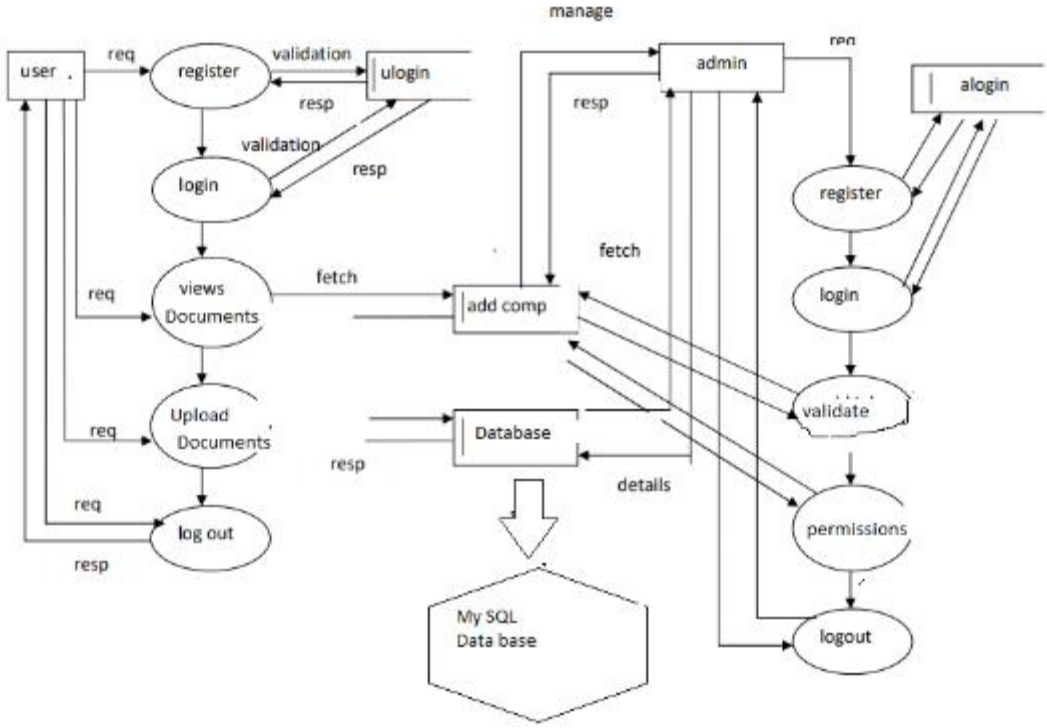
Incremental integration is the antithesis of the big bang approach. The program is constructed and tested in small increments, where errors are easier to isolate and correct, interfaces are more likely to be tested completely, and a systematic test approach may be applied.

4.2.7 System Testing

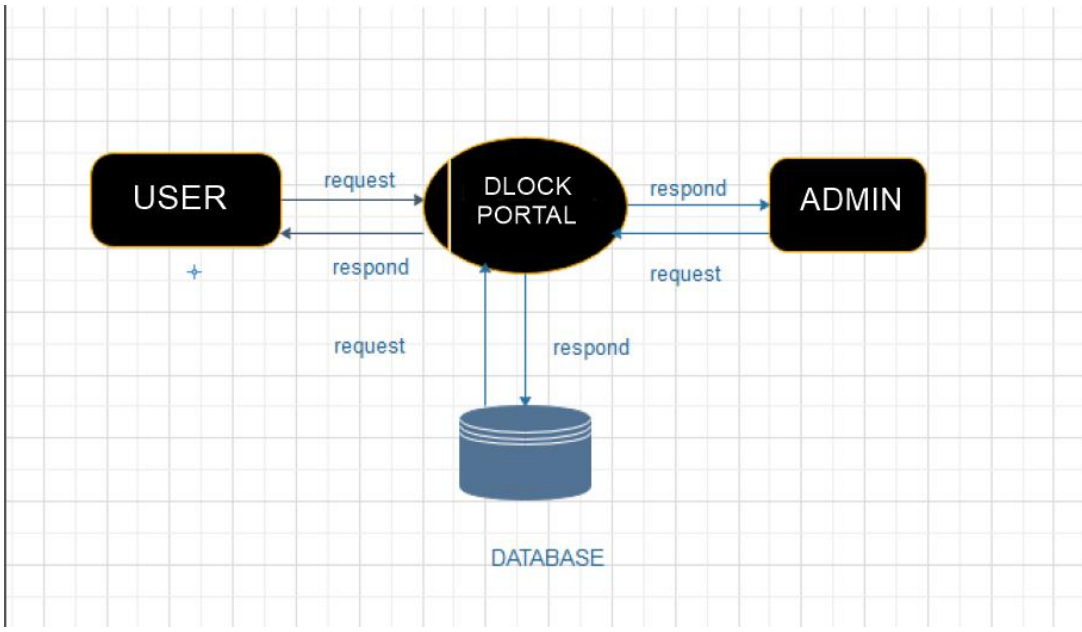
System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

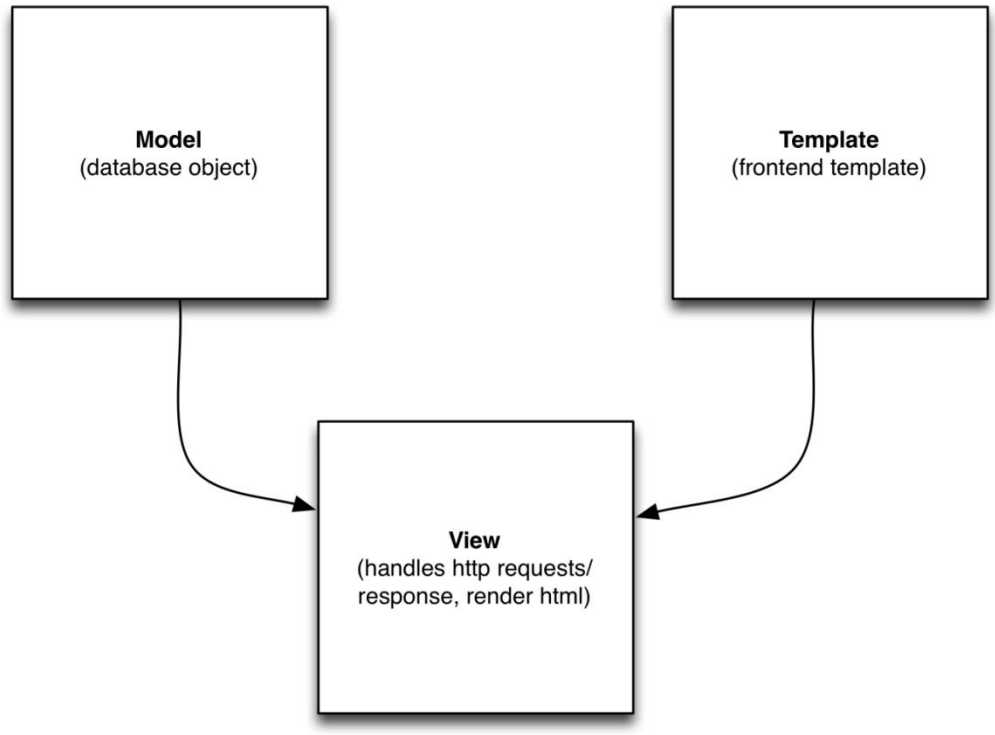
4.3 E-R Diagram

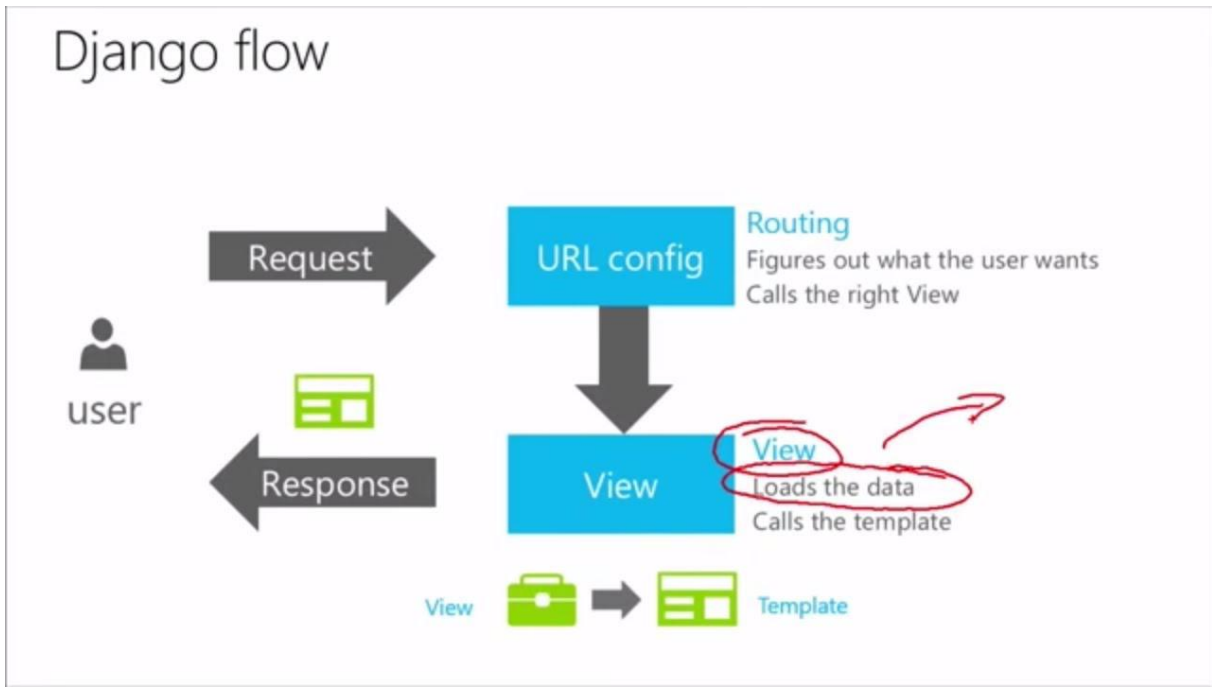




4.4 DFD(Data flow Diagram)




Model
(database object)
Template
(frontend template)
View
(handles http requests/
response, render html)


Django flow
Request
URL config
Routing
Figures out what the user wants
Calls the right View
user
Response
View
View
Loads the data
Calls the template
View
Template

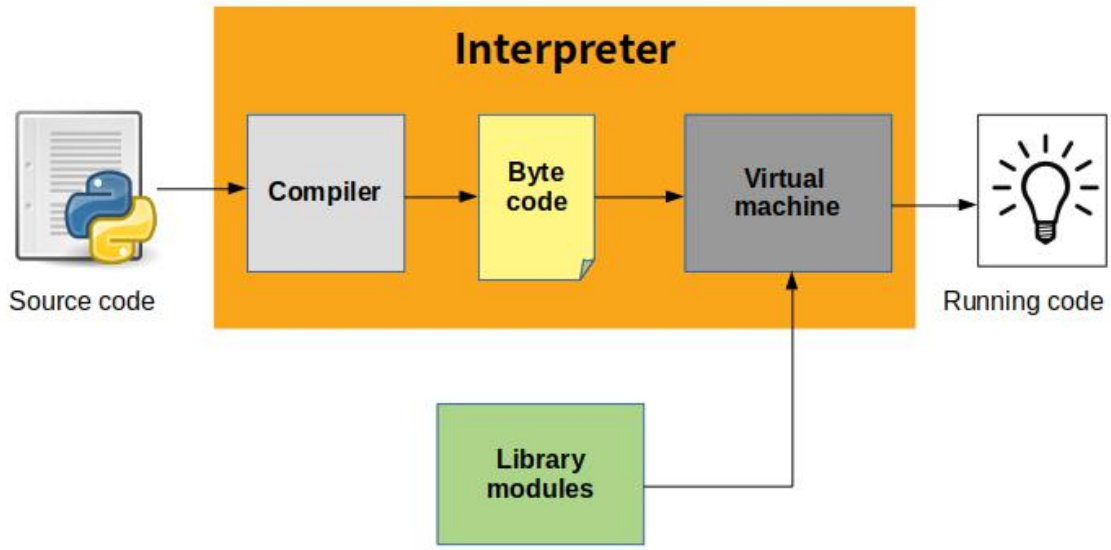
5. **SELECTED SOFTWARE**

5.1 PYTHON INTERPRETER

5.1.1 Introduction

An interpreter is a **program that reads and executes code**. This includes source code, pre-compiled code, and scripts. Common interpreters include Perl, Python, and Ruby interpreters, which execute Perl, Python, and Ruby code respectively.

Python is an **interpreted language**. **Interpreted languages** do not need to be compiled to run. A program called an interpreter runs **Python** code on almost any kind of computer. This means that a programmer can change the code and quickly see the results.



5.1.2 Python Interpreter & its Environment (Source Code Encoding)

The default encoding for a Python source file is UTF-8. This is a Unicode Standard variable-width character encoding; it can encode 1,112,064 valid code points in Unicode using up to four 8-bit bytes. Using this encoding, we can use characters of most languages – we can use these in string literals, comments, and identifiers. Since the standard library makes use of ASCII characters only, you must declare the use of this encoding to your editor. This is to ensure that all such characters display without a problem. The font should be such that supports all characters in the file.
We add this comment as the first line of the file we want to use it in-

```
1. # -*- coding: encoding -*-
```

In this, encoding is a valid codec that Python supports. Similarly, when you want to use the Windows-1252 encoding, you can use this as the first line of code:

```
1. # -*- coding: cp1252 -*-
```

However, when you want to begin code with a UNIX shebang line, you can put the comment for encoding second-

```
1. #!/usr/bin/env python3
2. # -*- coding: cp1252 -*-
```

➢ How to Invoke the Python Interpreter?

On your machine, you can find your interpreter at an address like:
C:\Python36
Or it may reside on the location you selected at the time of installation. Add path using this command:

```
1. set path=%path%;C:\python36
```

a. Start the Python Interpreter

On Windows, when you want to run the Python interpreter in the shell, you can type the following:

```
1. $python
```

To get out of the interpreter in disassembling the Bytecode shell, you can type:

```
1. >>> quit()
```

b. Features of Python Interpreter

Python interpreter offers some pretty cool features:

- Interactive editing

- History substitution
- Code completion on systems with support for readline

In the first Python prompt, try pressing the following keys:

Ctrl+P

This tells you if your interpreter supports command-line editing. A beep indicates that it does support command-line editing. Otherwise, it will either perform a no-operation or echo ^*p* to indicate it isn't available.

c. Passing Arguments

When you pass a script name and additional arguments to the shell when invoking the Python interpreter, it turns these into a list of strings. Then, it assigns these to the variable *argv* in the sys module. The following command will give us a list of this-

```
1. import sys
```

Without a script or arguments, sys.argv[0] denotes an empty string. A script name of '-' means that it sets sys.argv[0] to '-', and with '*-c*', it is set to '*-c*'. A value of '*-m*' sets sys.argv[0] to the module's full name. The command/ module handles the options after '*-c*' or '*-m*'.

d. Interactive Mode

Python interpreter is in an interactive mode when it reads commands from a tty. The primary prompt is the following:

```
1. >>>
```

When it shows this prompt, it means it prompts the developer for the next command. This is the REPL. Before it prints the first prompt, Python interpreter prints a welcome message that also states its version number and a copyright notice.
This is the secondary prompt:

```
1. …
```

This prompt denotes continuation lines.

```
1. $ python3.7
2. Python 3.7 (default, Jul 16 2018, 04:38:07)
3. [GCC 4.8.2] on Windows
4. Type "help", "copyright", "credits" or "license" for more information.
5. >>>
```

You will find continuation lines when working with a multi-line construct:

```
1. >>> it_rains =True
2. >>> if it_rains:
3. print("The produce will be good")
```

The produce will be good

You can also use the Python interpreter as a calculator:

```
1. >>> 2*7
2. 14
3. >>> 4/2
4. 2.0
```

➢ How Does Python Interpreter Works?

Well, internally, four things happen in a REPL:

i. Lexing- The lexer breaks the line of code into tokens.

ii. Parsing- The parser uses these tokens to generate a structure, here, an Abstract Syntax Tree, to depict the relationship between these tokens.

iii. Compiling- The compiler turns this AST into code object(s).

iv. Interpreting- The interpreter executes each code object.

➢ Does Python Interpreter Works?

a. Function Objects & Code Objects

When we talk of function objects, we mean to say that in Python, functions are first-class objects (functions indeed are objects). You can pass them around and talk about them without making a call to them.

```
1. >>> def bar(a):
2. x=3
3. return x+a
4. >>> bar
```

<function bar at 0x107ef7aa2>

Now bar.__code__ returns a code object:

1. >>> bar.__code__
2. <code object bar at 0x107eeccb2, file "<stdin>", line 1>

So, we conclude that a code object is an attribute of a function object. The dir() function will tell us more about the function:

1. >>> dir(bar.__code__)

['__class__', '__cmp__', '__delattr__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__gt__', '__hash__', '__init__', '__le__', '__lt__', '__ne__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__',
'__subclasshook__', 'co_argcount', 'co_cellvars', 'co_code', 'co_consts', 'co_filename',
'co_firstlineno', 'co_flags', 'co_freevars', 'co_lnotab', 'co_name', 'co_names', 'co_nlocals',
'co_stacksize', 'co_varnames']

This gives us the attributes of the code object. Values of some more attributes:

1. >>> bar.__code__.co_varnames

('a', 'x')

1. >>> bar.__code__.co_consts
2. **(None, 3)**

1. >>> bar.__code__.co_argcount

b. The Bytecode

The following command gives us the bytecode:

1. >>> bar.__code__.co_code

'd\x01\x00}\x01\x00|\x01\x00|\x00\x00\x17S'

This is a series of bytes, each of which the interpreter loops through and then makes an execution.

c. Disassembling the Bytecode

We will use the dis() method from the dis module to understand what's going on- this isn't part of what the interpreter does.

1. >>>import dis
2. >>> dis.dis(bar.__code__)

2 0 LOAD_CONST 1 (3)
3 STORE_FAST 1 (x)
3 6 LOAD_FAST 1 (x)
9 LOAD_FAST 0 (a)
12BINARY_ADD
13RETURN_VALUE

In this, the first set of numbers is the line numbers in the actual code. The second one depicts offsets of the bytecode. Then comes the set of names for the bytes- for human readability. The next column depicts the arguments and the last column lists the constants and names in the fourth column.

1. >>> bar.__code__.co_consts[1]

1. >>> bar.__code__.co_varnames[1]

'x'

5. Conclusion

Hence, we can say the compiler for Python generates bytecode for the interpreter. The Python interpreter uses this with the virtual machine. The same bytecode doesn't always end up doing the same things. This is another thing that makes Python dynamic. Also, the default prompt for the interpreter is >>>.

5.1.3 Python

what is Python?

Chances you are asking yourself this. You may have found this book because you want to learn to program but don't know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to "big name" languages. Hopefully I can explain it for you.

Python concepts

If your not interested in the the hows and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it's a great one to start programming with.

• Open source general-purpose language.

• Object Oriented, Procedural, Functional

• Easy to interface with C/ObjC/Java/Fortran

• Easy-ish to interface with C++ (via SWIG)

• Great interactive environment

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 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

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol- 68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

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.

Dynamic vs Static

Types Python is a dynamic-typed language. Many other languages are static typed, Such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of “thing” each data value is.

For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a “float” type.

This tells the compiler that the only data that can be used for that variable must be a floating point number, i.e. a number with a decimal point.

If any other data value was assigned to that variable, the compiler would give an error when trying to compile the program.

Python, however, doesn’t require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating point number) you need in your program.

With a static typed language, you have to decide the memory size the variable can take when you first initialize that variable. A double is a floating point value that can handle a much larger number than a normal float (the actual memory sizes depend on the operating environment).

If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double.

With Python, it doesn’t matter. You simply give it whatever number you want and Python will take care of manipulating it as needed. It even works for derived values.

For example, say you are dividing two numbers. One is a floating point number and one is an integer. Python realizes that it’s more accurate to keep track of decimals so it automatically calculates the result as a floating point number

Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types −

- Numbers
- String
- List
- Tuple
- Dictionary

Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them

Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The

plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as **read-only** lists.

Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Different modes in python

Python has two basic modes: normal and interactive.

The normal mode is the mode where the scripted and finished .py files are run in the Python interpreter.

Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole

20 Python libraries

1. Requests. The most famous http library written by kenneth reitz. It's a must have for every python developer.

2. Scrapy. If you are involved in webscraping then this is a must have library for you. After using this library you won't use any other.

3. wxPython. A gui toolkit for python. I have primarily used it in place of tkinter. You will really love it.

4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.

5. SQLAlchemy. A database library. Many love it and many hate it. The choice is yours.

6. BeautifulSoup. I know it's slow but this xml and html parsing library is very useful for beginners.

7. Twisted. The most important tool for any network application developer. It has a very beautiful api and is used by a lot of famous python developers.

8. NumPy. How can we leave this very important library ? It provides some advance math functionalities to python.

9. SciPy. When we talk about NumPy then we have to talk about scipy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.

11. Pygame. Which developer does not like to play games and develop them ? This library will help you achieve your goal of 2d game development.

12. Pyglet. A 3d animation and game creation engine. This is the engine in which the famous python port of minecraft was made

13. pyQT. A GUI toolkit for python. It is my second choice after wxpython for developing GUI's for my python scripts.

14. pyGtk. Another python GUI library. It is the same library in which the famous Bittorrent client is created.

15. Scapy. A packet sniffer and analyzer for python made in python.

16. pywin32. A python library which provides some useful methods and classes for interacting with windows.

17. nltk. Natural Language Toolkit – I realize most people won't be using this one, but it's generic enough. It is a very useful library if you want to manipulate strings. But it's capacity is beyond that. Do check it out.

18. nose. A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.

19. SymPy. SymPy can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.

20. IPython. I just can't stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

Python class and objects

These are the building blocks of OOP. class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are very powerful and currently "the big thing" in most programming languages. Hence, there are several chapters dedicated to OOP later in the book.

The class is the most basic component of object-oriented programming. Previously, you learned how to use functions to make your program do something.

Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects.

When I first learned C and C++, I did great; functions just made sense for me.

Having messed around with BASIC in the early '90s, I realized functions were just like subroutines so there wasn't much new to learn.

However, when my C++ course started talking about objects, classes, and all the new features of OOP, my grades definitely suffered.

Once you learn OOP, you'll realize that it's actually a pretty powerful tool. Plus many Python libraries and APIs use classes, so you should at least be able to understand what the code is doing.

One thing to note about Python and OOP: it's not mandatory to use objects in your code in a way that works best; maybe you don't need to have a full-blown class with initialization code and methods to just return a calculation. With Python, you can get as technical as you want.

As you've already seen, Python can do just fine with functions. Unlike languages such as Java, you aren't tied down to a single way of doing things; you can mix functions and classes as necessary in the same program. This lets you build the code

Objects are an encapsulation of variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially a template to create your objects.

Here’s a brief list of Python OOP ideas:

• The class statement creates a class object and gives it a name. This creates a new namespace.

• Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: ClassName.Attribute.

• Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.

• Calling a class (just like a function) creates a new instance of the class. This is where the multiple copies part comes in.

• Each instance gets ("inherits") the default class attributes and gets its own namespace. This prevents instance objects from overlapping and confusing the program.

• Using the term self identifies a particular instance, allowing for per-instance attributes. This allows items such as variables to be associated with a particular instance.

Inheritance

First off, classes allow you to modify a program without really making changes to it.

To elaborate, by subclassing a class, you can change the behavior of the program by simply adding new components to it rather than rewriting the existing components.

As we’ve seen, an instance of a class inherits the attributes of that class.

However, classes can also inherit attributes from other classes. Hence, a subclass inherits from a superclass allowing you to make a generic superclass that is specialized via subclasses.

The subclasses can override the logic in a superclass, allowing you to change the behavior of your classes without changing the superclass at all.

Operator Overloads

Operator overloading simply means that objects that you create from classes can respond to actions (operations) that are already defined within Python, such as addition, slicing, printing, etc.

Even though these actions can be implemented via class methods, using overloading ties the behavior closer to Python’s object model and the object interfaces are more consistent to Python’s built-in objects, hence overloading is easier to learn and use.

User-made classes can override nearly all of Python’s built-in operation methods.

Exceptions

I've talked about exceptions before but now I will talk about them in depth. Essentially, exceptions are events that modify program's flow, either intentionally or due to errors.

They are special events that can occur due to an error, e.g. trying to open a file that doesn't exist, or when the program reaches a marker, such as the completion of a loop.

Exceptions, by definition, don't occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python.

Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

• Accessing a non−existent dictionary key will raise a KeyError exception.

• Searching a list for a non−existent value will raise a ValueError exception

• Calling a non−existent method will raise an AttributeError exception.

• Referencing a non−existent variable will raise a NameError exception.

• Mixing datatypes without coercion will raise a TypeError exception.

One use of exceptions is to catch a fault and allow the program to continue working; we have seen this before when we talked about files.

This is the most common way to use exceptions. When programming with the Python command line interpreter, you don't need to worry about catching exceptions.

Your program is usually short enough to not be hurt too much if an exception occurs.

Plus, having the exception occur at the command line is a quick and easy way to tell if your code logic has a problem.

However, if the same error occurred in your real program, it will fail and stop working. Exceptions can be created manually in the code by raising an exception.

It operates exactly as a system-caused exceptions, except that the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions is that, by their nature, they don't put any overhead on the code processing. Because exceptions aren't supposed to happen very often, they aren't processed until they occur.

Exceptions can be thought of as a special form of the if/elif statements. You can realistically do the same thing with if blocks as you can with exceptions. However, as

already mentioned, exceptions aren't processed until they occur; if blocks are processed all the time.

Proper use of exceptions can help the performance of your program. The more infrequent the error might occur, the better off you are to use exceptions; using if blocks requires Python to always test extra conditions before continuing. Exceptions also make code management easier: if your programming logic is mixed in with error-handling if statements, it can be difficult to read, modify, and debug your program.

User-Defined Exceptions

I won't spend too much time talking about this, but Python does allow for a programmer to create his own exceptions.

You probably won't have to do this very often but it's nice to have the option when necessary.

However, before making your own exceptions, make sure there isn't one of the built-in exceptions that will work for you.

They have been "tested by fire" over the years and not only work effectively, they have been optimized for performance and are bug-free.

Making your own exceptions involves object-oriented programming, which will be covered in the next chapter

To make a custom exception, the programmer determines which base exception to use as the class to inherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class.

To make a custom exception, simply inherit the base exception and define what it will do.

Python modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library. To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a *module*; definitions from a module can be *imported* into other modules or into the *main* module.

Testing code

As indicated above, code is usually developed in a file using an editor. To test the code, import it into a Python session and try to run it. Usually there is an error, so you go back to the file, make a correction, and test again.

This process is repeated until you are satisfied that the code works. T he entire process is known as the development cycle.

There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid.

This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error

Functions in Python

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task.

To carry out that specific task, the function might or might not need multiple inputs. When the task is carried out, the function can or can not return one or more values.There are three types of functions in python: help() ,min(), print().

Python Namespace

Generally speaking, a namespace (sometimes also called a context) is a naming system for making names unique to avoid ambiguity. Everybody knows a namespacing system from daily life, i.e. the naming of people in firstname and familiy name (surname). An example is a network: each network device (workstation, server, printer, ...) needs a unique name and address. Yet another example is the directory structure of file systems. The same file name can be used in different directories, the files can be uniquely accessed via the pathnames. Many programming languages use namespaces or contexts for identifiers. An identifier defined in a namespace is associated with that namespace.

This way, the same identifier can be independently defined in multiple namespaces. (Like the same file names in different directories) Programming languages, which support namespaces, may have different rules that determine to which namespace an identifier belongs.

Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces. A

Some namespaces in Python:

- **global names** of a module
- **local names** in a function or method invocation
- **built-in names**: this namespace contains built-in functions (e.g. abs(), cmp(), ...) and built-in exception names

Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

Python Web Frameworks

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make 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.

Common web framework functionality

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing

2. HTML, XML, JSON, and other output format templating

3. Database manipulation

4. Security against Cross-site request forgery (CSRF) and other attacks

5. Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

Comparing web frameworks

There is also a repository called compare-python-web-frameworks where the same web application is being coded with varying Python web frameworks, templating engines and object.

Web framework resources

- When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.

- Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.

- what is a web framework? is an in-depth explanation of what web frameworks are and their relation to web servers.

- Django vs Flash vs Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.

- This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.

- Python’s web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as

well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.

- What web frameworks do you use and why are they awesome? is a language agnostic Reddit discussion on web frameworks. It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
- This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?". The votes aren't as important as the list of the many frameworks that are available to Python developers.

Web frameworks learning checklist

1. Choose a major Python web framework (Django or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.

2. Work through a detailed tutorial found within the resources links on the framework's page.

3. Study open source examples built with your framework of choice so you can take parts of those projects and reuse the code in your application.

4. Build the first simple iteration of your web application then go to the deployment section to make it accessible on the web.

Python-Data Base Communication

Connector/Python provides a connect() call used to establish connections to the MySQL server. The following sections describe the permitted arguments for connect() and describe how to use option files that supply additional arguments.

A database is an organized collection of data. The data are typically organized to model aspects of reality in a way that supports processes requiring this information.

The term "database" can both refer to the data themselves or to the database management system. The Database management system is a software application for the interaction between users database itself.

Databases are popular for many applications, especially for use with web applications or customer-oriented programs

Users don't have to be human users. They can be other programs and applications as well. We will learn how Python or better a Python program can interact as a user of an SQL database.

This is an introduction into using SQLite and MySQL from Python.

The Python standard for database interfaces is the Python DB-API, which is used by Python's database interfaces.

The DB-API has been defined as a common interface, which can be used to access relational databases.

In other words, the code in Python for communicating with a database should be the same, regardless of the database and the database module used. Even though we use lots of SQL examples, this is not an introduction into SQL but a tutorial on the Python interface.

SQLite is a simple relational database system, which saves its data in regular data files or even in the internal memory of the computer, i.e. the RAM.

It was developped for embedded applications, like Mozilla-Firefox (Bookmarks), Symbian OS or Android.

SQLITE is "quite" fast, even though it uses a simple file. It can be used for large databases as well.

If you want to use SQLite, you have to import the module sqlite3. To use a database, you have to create first a Connection object.

The connection object will represent the database. The argument of connection - in the following example "companys.db" - functions both as the name of the file, where the data will be stored, and as the name of the database. If a file with this name exists, it will be opened.

It has to be a SQLite database file of course! In the following example, we will open a database called company.

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library. For notes detailing the changes in each release of Connector/Python, see MySQL Connector/Python Release Notes.

MySQL Connector/Python includes support for:

- Almost all features provided by MySQL Server up to and including MySQL Server version 5.7.
- Converting parameter values back and forth between Python and MySQL data types, for example Python datetime and MySQL DATETIME. You can turn automatic conversion on for convenience, or off for optimal performance.

- All MySQL extensions to standard SQL syntax Protocol compression, which enables compressing the data stream between the client and server.
- Connections using TCP/IP sockets and on Unix using Unix sockets.
- Secure TCP/IP connections using SSL.
- Self-contained driver. Connector/Python does not require the MySQL client library

5.2 DJANGO

Introduction

Django is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because Django provides built-in features for everything including Django Admin Interface, default database – SQLlite3, etc. When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use and that too for rapid development.

Django is a Python-based web framework which allows you to quickly create web application without all of the installation or dependency problems that you normally will find with other frameworks.
When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use.

Why use Django?

The following is a nonexhaustive list of the advantages of using Django:

- Django is published under the BSD license, which assures that web applications can be used and modified freely without any problems; it's also free.
- Django is fully customizable. Developers can adapt to it easily by creating modules or overridden framework methods.
- This modularity adds other advantages. There are a lot of Django modules that you can integrate into Django. You can get some help with other people's because you will often find high-quality modules that you might need.
- Using Python in this framework allows you to have benefits from all Python libraries and assures a very good readability.
- Django is a framework whose main goal is perfection. It was specifically made for people who want clear code and a good architecture for their applications. It totally respects the **Don't Repeat Yourself** (**DRY**) philosophy, which means keeping the code simple without having to copy/paste the same parts in multiple places.
- With regards to quality, Django integrates lots of efficient ways to perform unit tests.
- Django is supported by a good community. This is a very important asset because it allows you to resolve issues and fix bugs very fast. Thanks to the community, we can also find code examples that show the best practices.

Django architecture

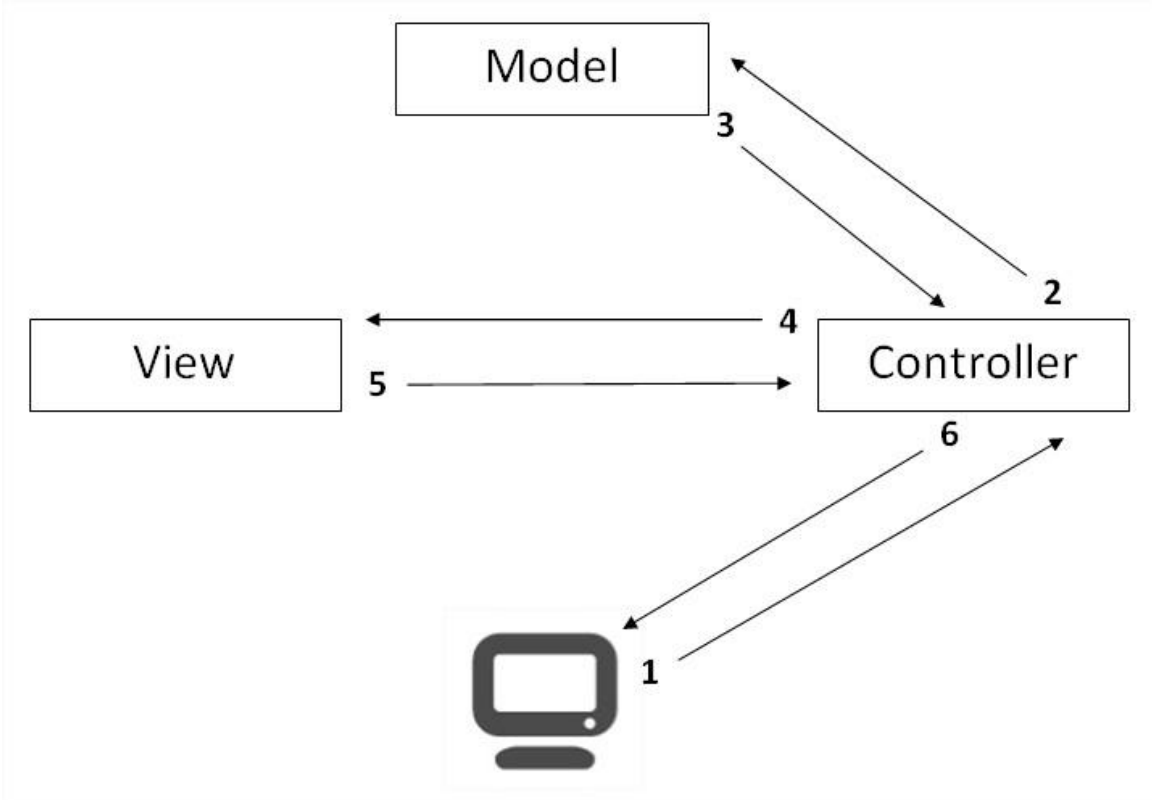
Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application.

MVT Structure has the following three parts –

Model: Model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySql, Postgres).

View: The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.

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.



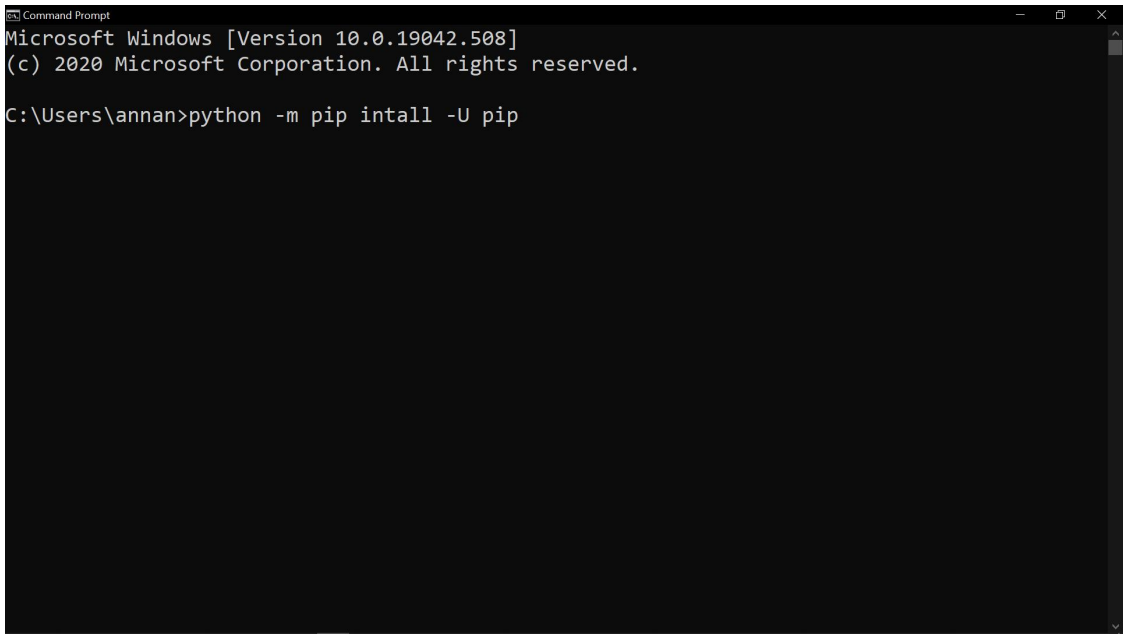
Installation of Django

Install python3 if not installed in your system (according to configuration of your system and OS) from here .

Note- Installation of Django in Linux and Mac is similar, here I am showing it in windows for Linux and mac just open terminal in place of command prompt and go through the following commands.

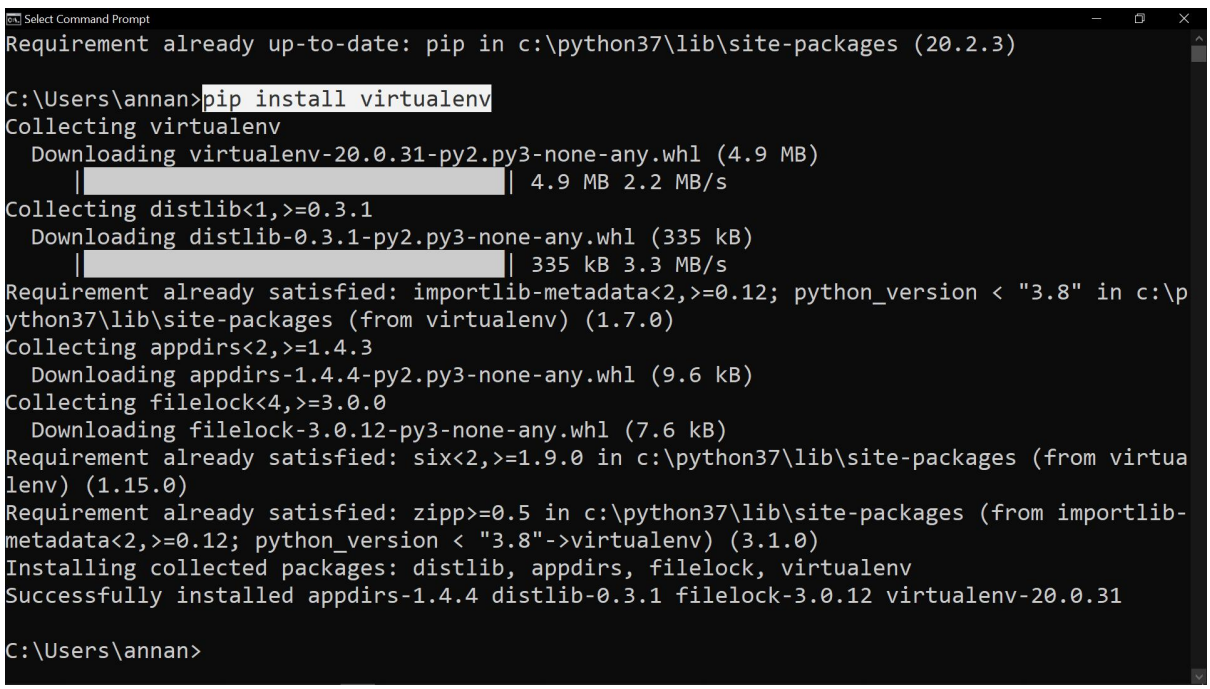
- **Install pip** - Open command prompt and enter following command-

  python -m pip install -U pip



- **Install virtual environment** - Enter following command in cmd-

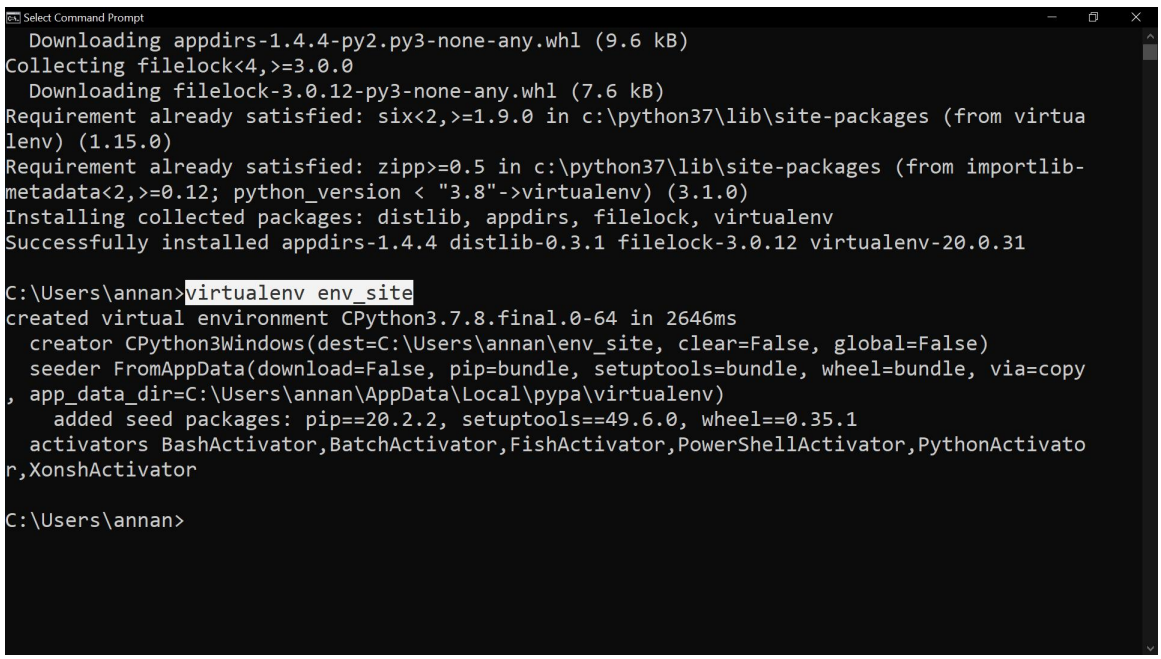
  pip install virtualenv



- **Set Virtual environment -** Setting up the virtual environment will allow you to edit the dependency which generally your system wouldn't allow.
  Follow these steps to set up a virtual environment-

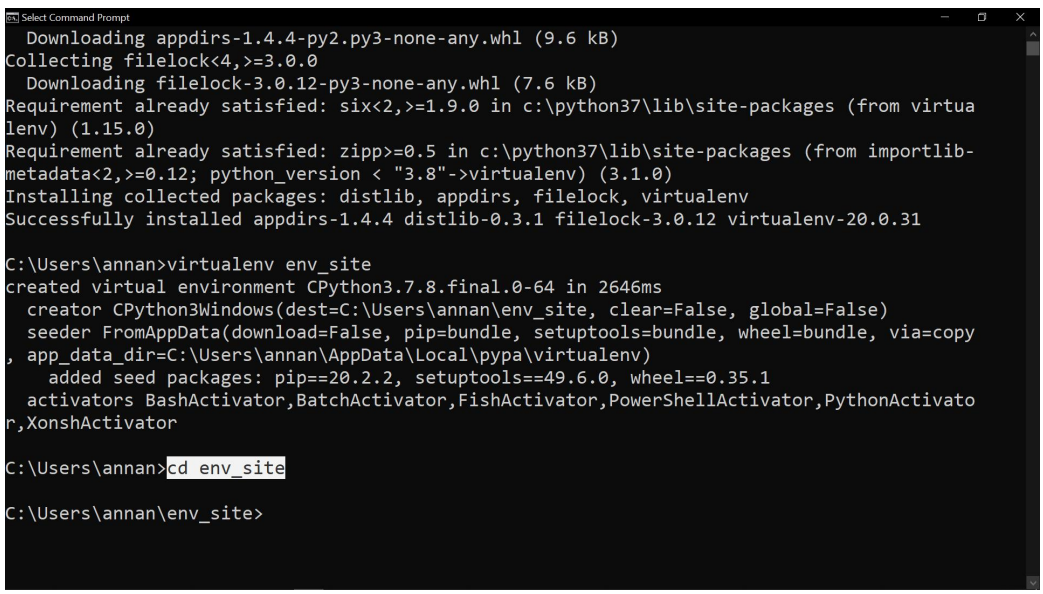
1. Create a virtual environment by giving this command in cmd-

virtualenv env_site



2. Change directory to env_site by this command-

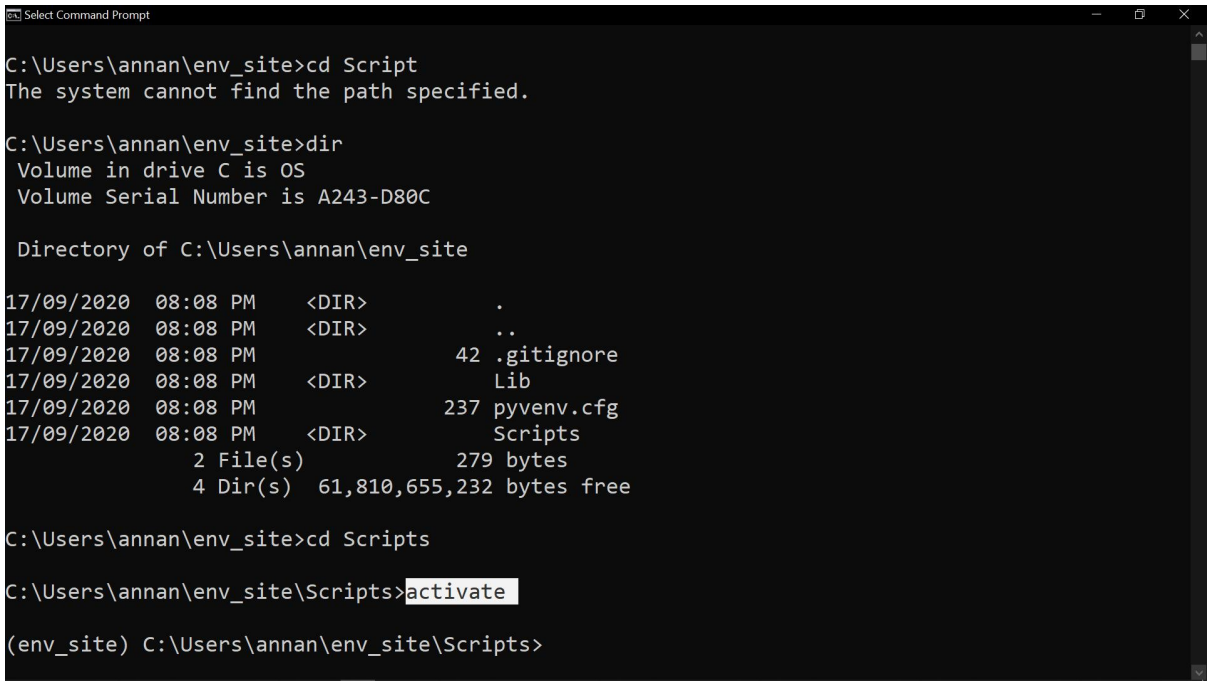
cd env_site



3. Go to Script directory inside env_site and activate virtual environment-

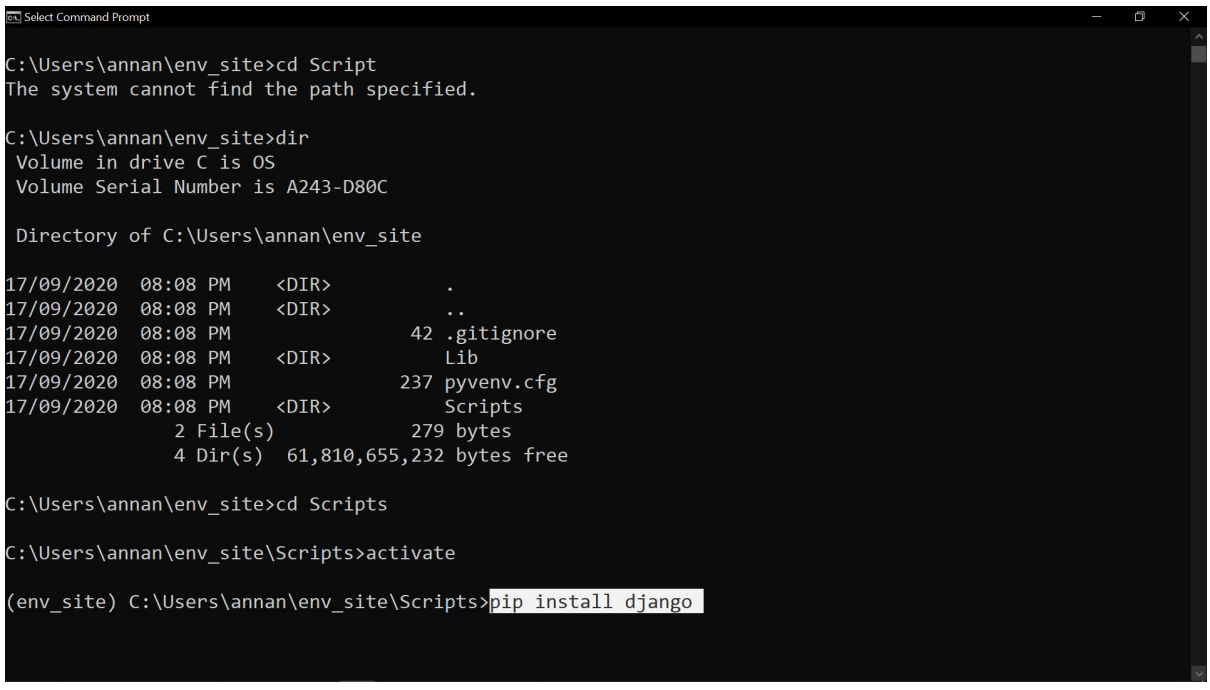
cd Script

activate



- **Install Django** - Install django by giving following command-

pip install django

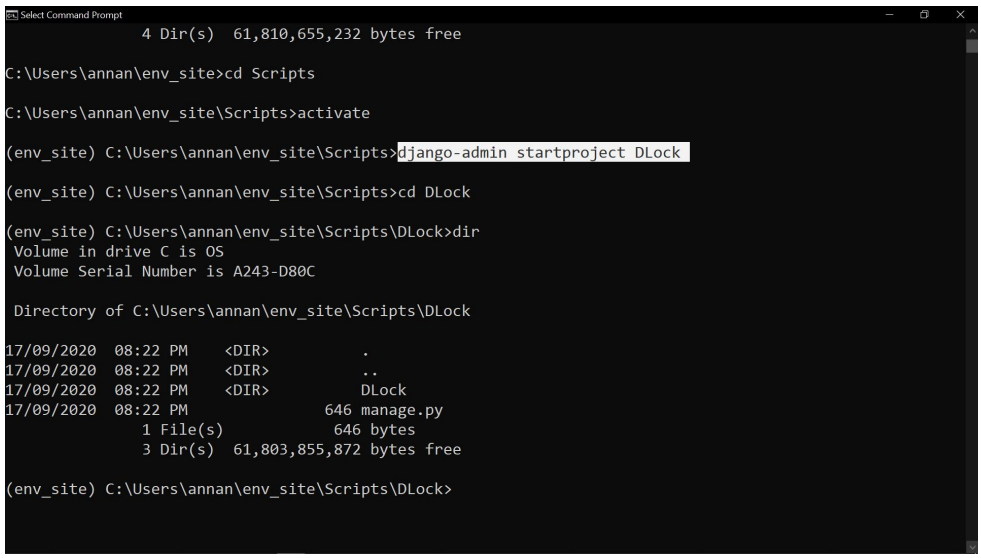


Creating a Project

Let's check how to create a basic project using Django after you have installed it in your pc.

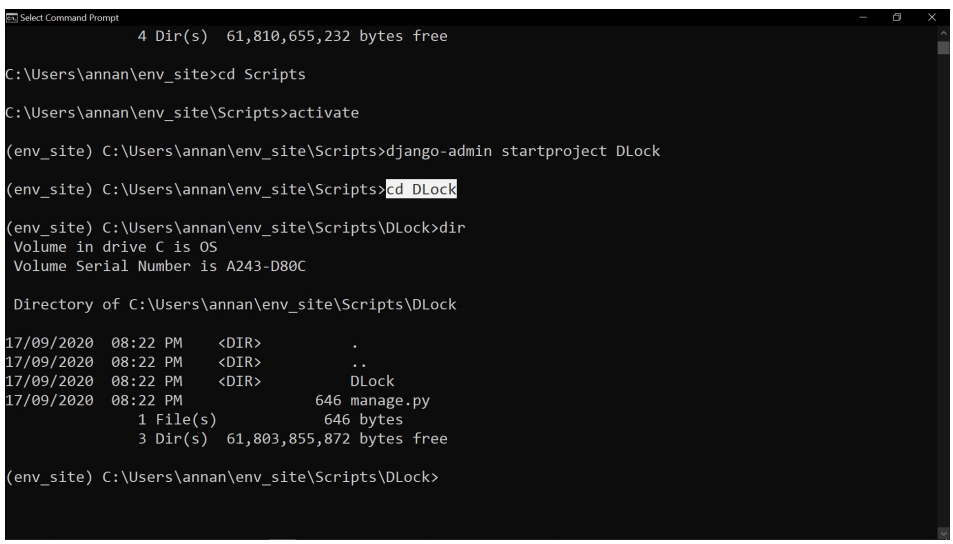
To initiate a project of Django on Your PC, open Terminal as I has used PyCharm editor for my project so I use Terminal present in PyCharm Editor, for windows users they can also use command prompt. Just set the path where you want to create the project folder and Enter the following command

Step 1. django-admin startproject projectName



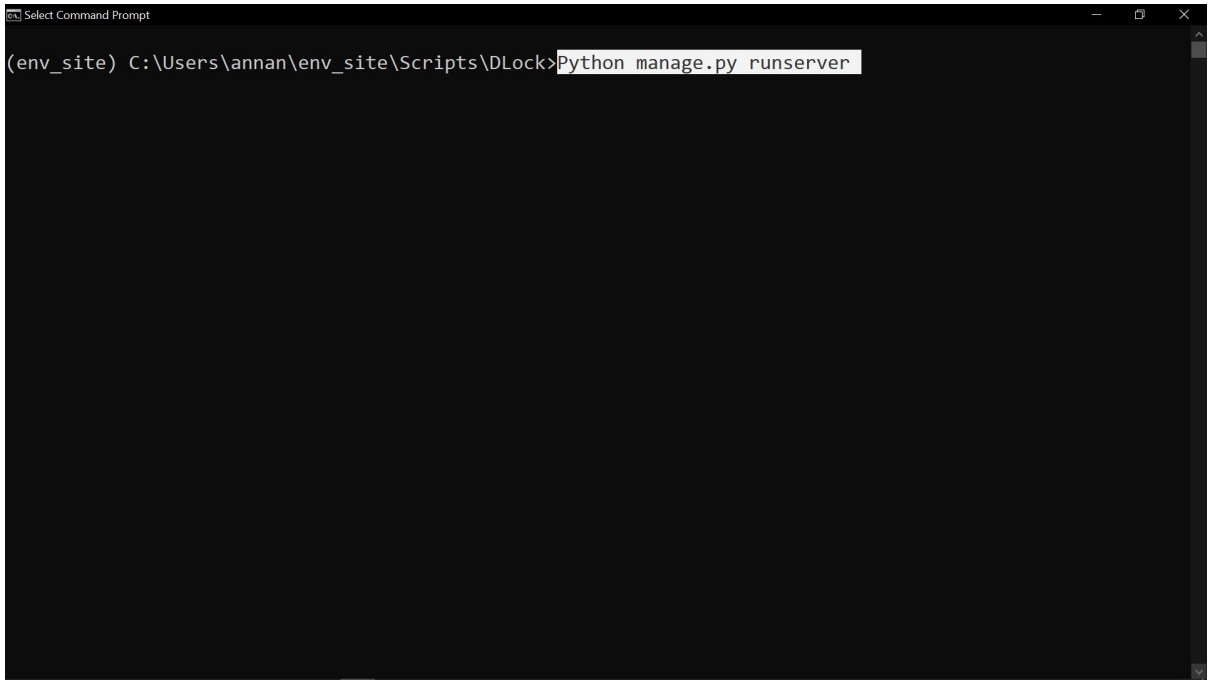
A New Folder with name projectName will be created. To enter in the project using terminal enter command

cd projectName

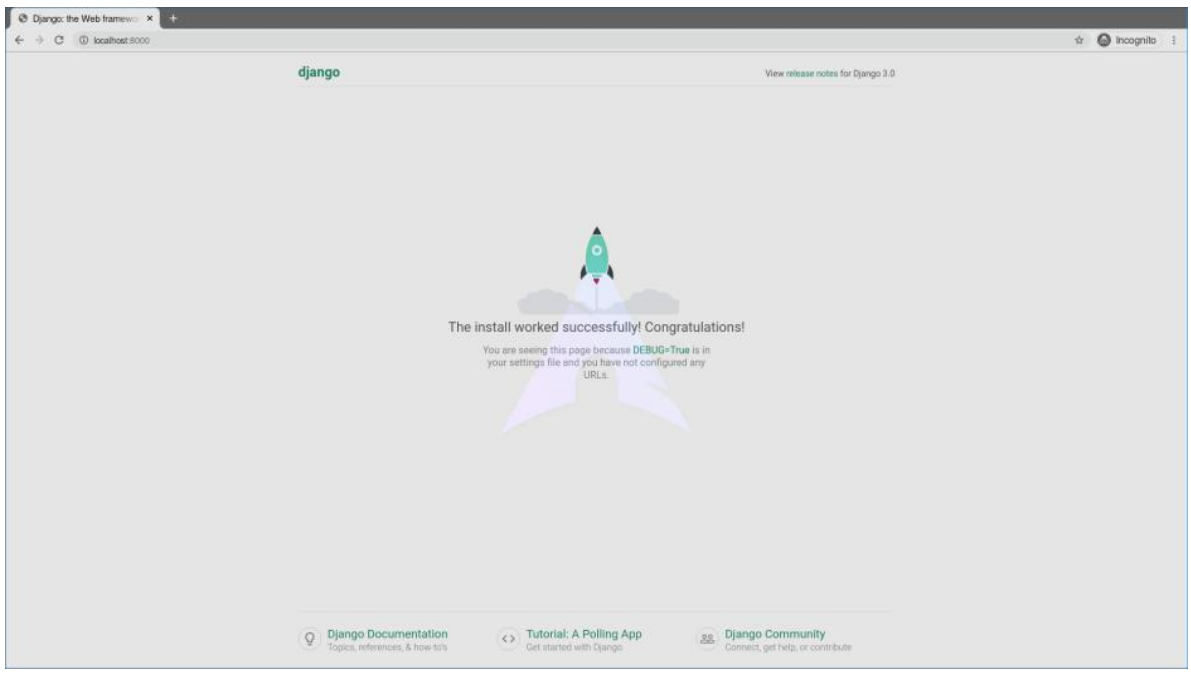


Now run,

Python manage.py runserver



Now visit http://localhost:8000/

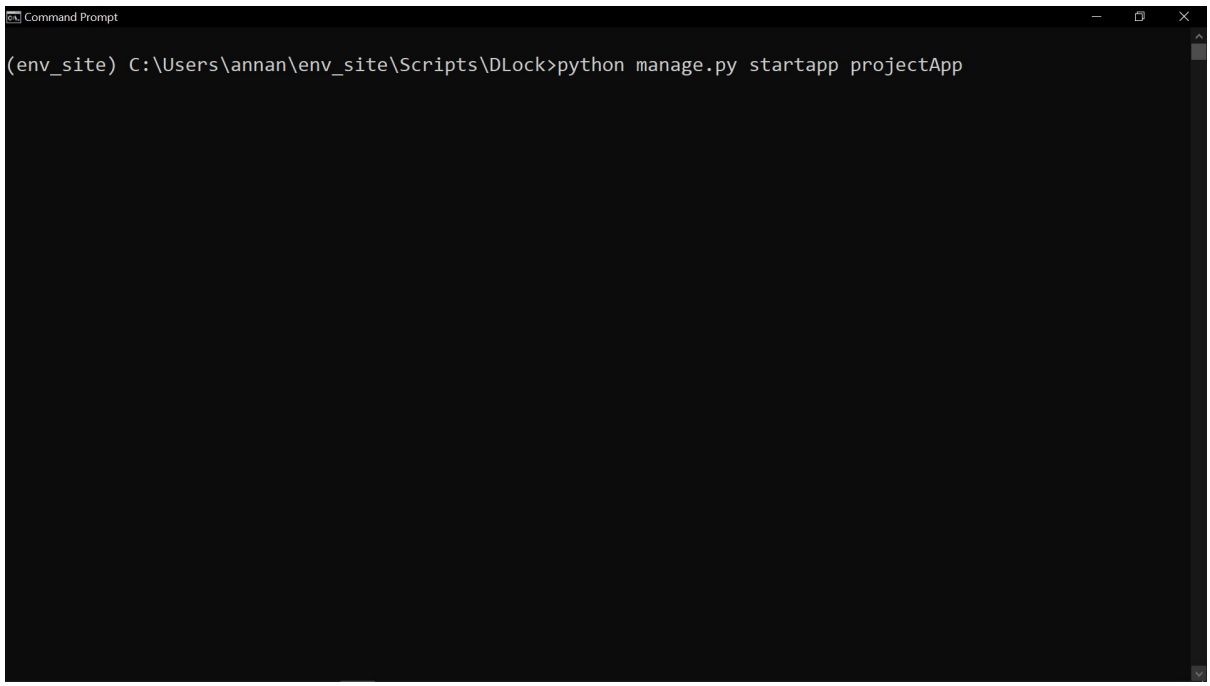


Creating an App

Django is famous for its unique and fully managed app structure. For every functionality, an app can be created like a completely independent module. This article will take you through how to create a basic app and add functionalities using that app.

- To create a basic app in your Django project you need to go to directory containing manage.py and from there enter the command :

python manage.py startapp projectApp



Now you can see your directory structure as under :

To consider the app in your project you need to specify your project name in INSTALLED_APPS list as follows in settings.py:

 # Application definition

INSTALLED_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

```
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'projectApp'
]
```

- So, we have finally created an app but to render the app using urls we need to include the app in our main project so that urls redirected to that app can be rendered. Let us explore it. Move to projectName-> projectName -> urls.py and add below code in the header

from django.urls import include

Now in the list of URL patterns, you need to specify app name for including your app urls. Here is the code for it –

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    # Enter the app name in following syntax for this to work
    path('', include("projectApp.urls")), ]
```

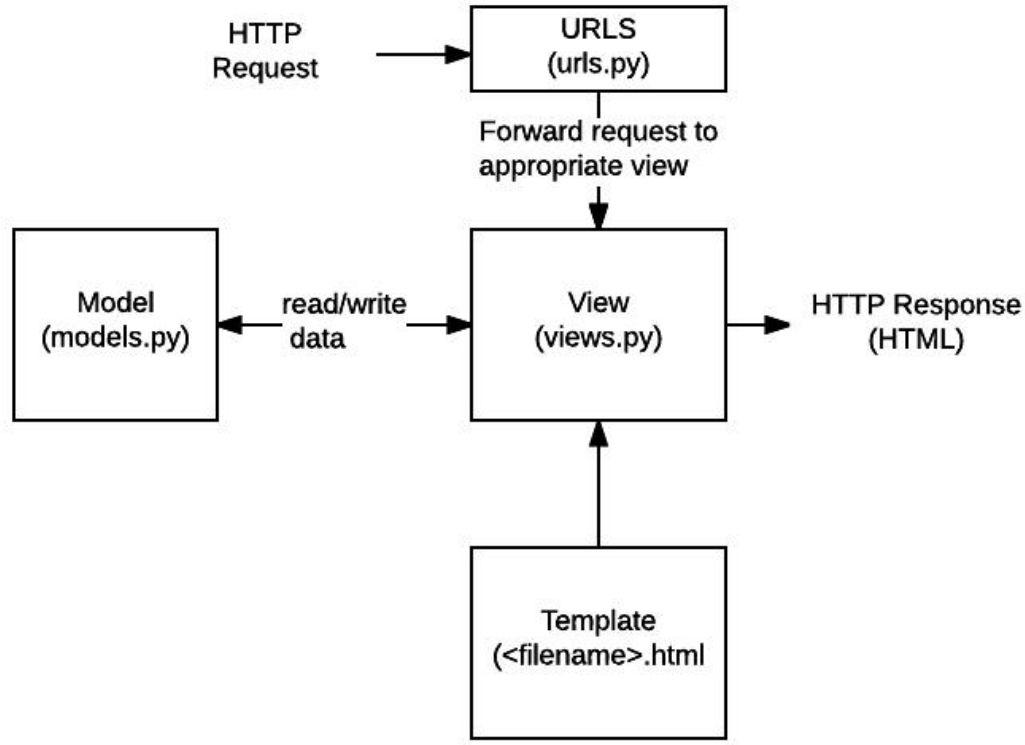
- Now You can use the default MVT model to create URLs, models, views, etc. in your app and they will be automatically included in your main project.

The main feature of Django Apps is independence, every app functions as an independent unit in supporting the main project.

What does Django code look like?

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or other client). When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template.

Django web applications typically group the code that handles each of these steps into separate files:



URLs: While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.

View: A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.

Models: Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.

Templates: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically create an HTML page using an HTML template, populating it with data from a model. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

Django Facilitate:

The preceding sections show the main features that you'll use in almost every web application: URL mapping, views, models and templates. Just a few of the other things provided by Django include:

Forms: HTML Forms are used to collect user data for processing on the server. Django simplifies form creation, validation, and processing.

User authentication and permissions: Django includes a robust user authentication and permission system that has been built with security in mind.

Caching: Creating content dynamically is much more computationally intensive (and slow) than serving static content. Django provides flexible caching so that you can store all or part of a rendered page so that it doesn't get re-rendered except when necessary.

Administration site: The Django administration site is included by default when you create an app using the basic skeleton. It makes it trivially easy to provide an admin page for site administrators to create, edit, and view any data models in your site.

Serialising data: Django makes it easy to serialise and serve your data as XML or JSON. This can be useful when creating a web service (a website that purely serves data to be consumed by other applications or sites, and doesn't display anything itself), or when creating a website in which the client-side code handles all the rendering of data.

Django Admin

The Django admin application can use your models to automatically build a site area that you can use to create, view, update, and delete records. This can save you a lot of time during development, making it very easy to test your models and get a feel for whether you have the right data. The admin application can also be useful for managing data in production, depending on the type of website. The Django project recommends it only for internal data management (i.e. just for use by admins, or people internal to your organization), as the model-centric approach is not necessarily the best possible interface for all users, and exposes a lot of unnecessary detail about the models.

All the configuration required to include the admin application in your website was done automatically when you created the skeleton project (for information about actual dependencies needed, see the Django docs here). As a result, all you must do to add your models to the admin application is to register them. At the end of this

article we'll provide a brief demonstration of how you might further configure the admin area to better display our model data.

After registering the models we'll show how to create a new "superuser", login to the site, and create some books, authors, book instances, and genres. These will be useful for testing the views and templates we'll start creating in the next tutorial.

Registering models

First, open admin.py in the catalog application (/locallibrary/catalog/admin.py). It currently looks like this — note that it already imports django.contrib.admin:

```
from django.contrib import admin

# Register your models here.
```

Register the models by copying the following text into the bottom of the file. This code simply imports the models and then calls admin.site.register to register each of them.

This is the simplest way of registering a model, or models, with the site. The admin site is highly customisable, and we'll talk more about the other ways of registering your models further down.

Creating a superuser

In order to log into the admin site, we need a user account with Staff status enabled. In order to view and create records we also need this user to have permissions to manage all our objects. You can create a "superuser" account that has full access to the site and all needed permissions using manage.py.

Call the following command, in the same directory as manage.py, to create the superuser. You will be prompted to enter a username, email address, and strong password.

python3 manage.py createsuperuser

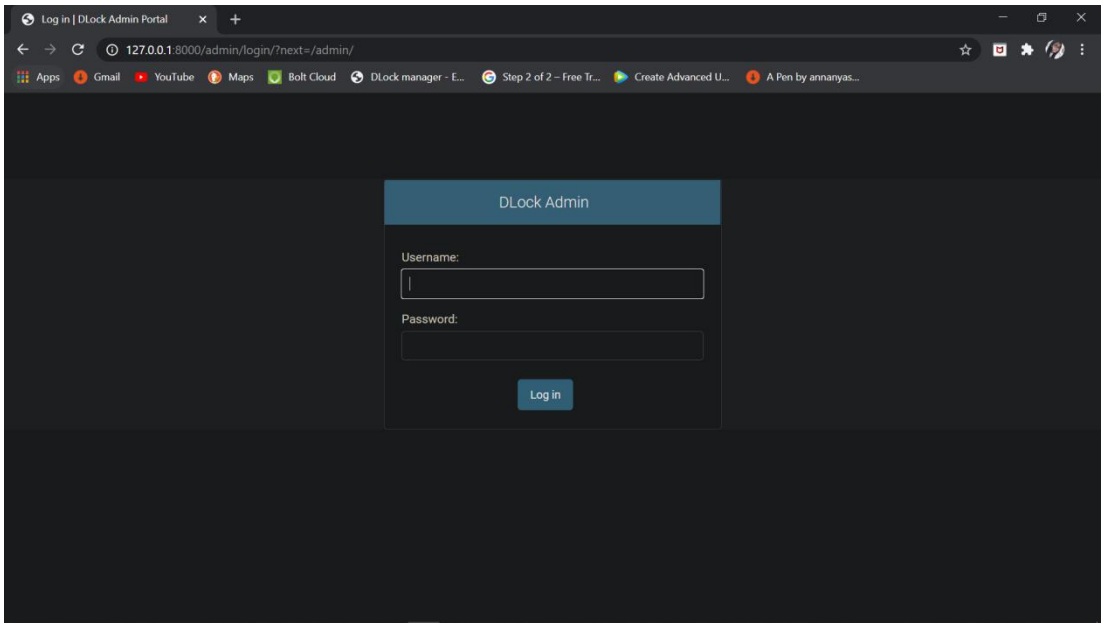
```
D:\persnal document\SampleProject\project\E-Docs\edocs>python manage.py createsuperuser
Username: adminsite
Email address: annanyasharma868@gmail.com
Password:
Password (again):
Superuser created successfully.

D:\persnal document\SampleProject\project\E-Docs\edocs>
```

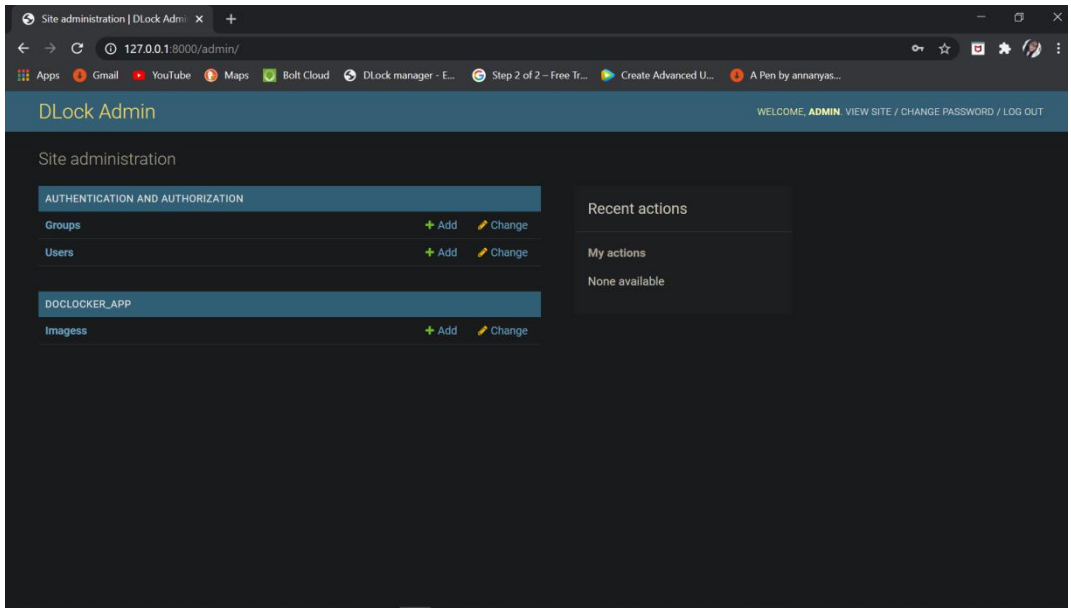
Once this command completes a new superuser will have been added to the database. Now restart the development server so we can test the login:

python3 manage.py runserver

Logging in and using the site

To login to the site, open the /admin URL (e.g. http://127.0.0.1:80s00/admin) and enter your new superuser userid and password credentials (you'll be redirected to the login page, and then back to the /admin URL after you've entered your details).





This part of the site displays all our models, grouped by installed application. You can click on a model name to go to a screen that lists all its associated records, and you can further click on those records to edit them. You can also directly click the Add link next to each model to start creating a record of that type.

5.3 PYCHARM

Introduction

PyCharm is a hybrid-platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE!

It supports two versions: v2.x and v3.x.

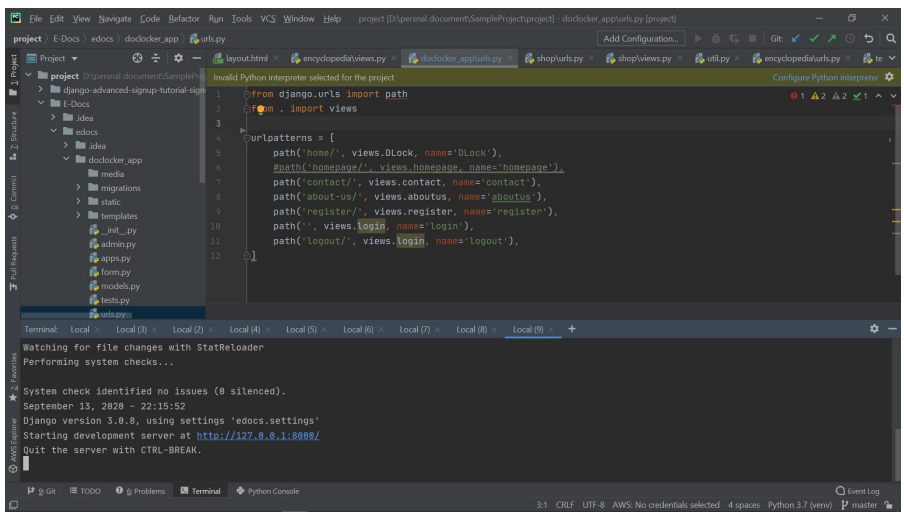
We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

What is an IDE?

To understand ‘What is PyCharm?’ and ‘What is PyCharm used for?,’ we should first be able to answer the question, ‘What is an IDE?’

An IDE consists of an editor and a compiler that we use to write and compile programs. It has a combination of features required for developing software.

The presence of an IDE makes the development process and programming much easier. It interprets what we are typing and suggests the relevant keyword to insert. We can distinguish between a class and a method as the IDE allocates different colors to them. The IDE also gives different colors for the right and the wrong keywords. If we are writing a wrong keyword, it tries to predict the keyword that we intend to write and auto completes it.#



Why should we use an IDE?

The major reasons for using an IDE for development are given below:

- An IDE consists of a text editor window where we can write our programs.
- It consists of a project editor window where we store all the necessary files of a software project.
- We can provide various inputs and check the efficiency of our program by inspecting the output we receive on the output window.
- If any error occurs, then the IDE shows warnings and suggestions on the output window so that we can resolve it.
- An IDE has a rack of modules and packages in one place that helps add features in our software applications.
- It helps increase efficiency in creating software.

Now, do you want to know which IDE we use for Python programming and application development? The most popular and widely used IDE for Python application development and programming is PyCharm.

Features of PyCharm: Why should we use it for our next Python project?

Below, we have compiled some of the essential features provided by PyCharm.

1. Intelligent Code Editor:

- It helps us write high-quality codes!
- It consists of color schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.
- It helps identify errors easily.
- It provides the autocomplete feature and instructions for the completion of the code.

2. Code Navigation:

- It helps developers in editing and enhancing the code with less effort and time.
- With code navigation, a developer can easily navigate to a function, class, or file.
- A programmer can locate an element, a symbol, or a variable in the source code within no time.

- Using the lens mode, further, a developer can thoroughly inspect and debug the entire source code.

3. Refactoring

- It has the advantage of making efficient and quick changes to both local and global variables.
- Refactoring in PyCharm enables developers to improve the internal structure without changing the external performance of the code.
- It also helps split up more extended classes and functions with the help of the extract method.
-

4. Assistance for Many Other Web Technologies:

- It helps developers create web applications in Python.
- It supports popular web technologies such as HTML, CSS, and JavaScript.
- Developers have the choice of live editing with this IDE. At the same time, they can preview the created/updated web page.
- The developers can follow the changes directly on a web browser.
- PyCharm also supports AnglularJS and NodeJS for developing web applications.

5. Support for Popular Python Web Frameworks

- PyCharm supports web frameworks such as Django.
- It provides the autocomplete feature and suggestions for the parameters of Django.
- It helps in debugging the codes of Django.
- It also assist web2py and Pyramid, the other popular web frameworks.

6. Assistance for Python Scientific Libraries

- PyCharm supports Python's scientific libraries such as Matplotlib, NumPy, and Anaconda.
- These scientific libraries help in building projects of Data Science and Machine Learning.
- It consists of interactive graphs that help developers understand data.

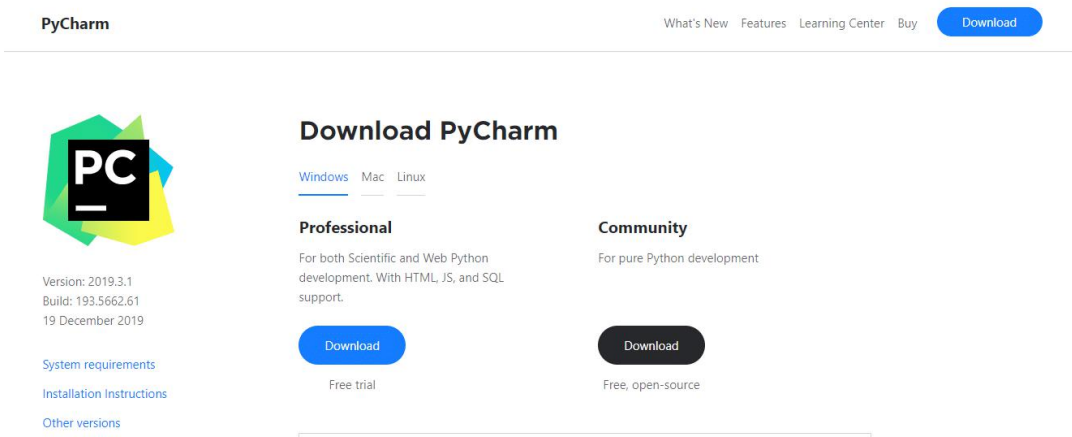
- It is capable of integrating with various tools such as IPython, Django, and Pytest. This integration helps innovate unique solutions.

Installing PyCharm

The installation of PyCharm requires no effort. You just need to open the official website with the following link:

www.jetbrains.com/pycharm/download/#section=windows

This link will open up the website as shown below:



There are two versions: **Professional** and **Community**. The Community version is free and open-source and the Professional version is the paid one. You can choose any according to your requirements.

5.4 Database - Sqlite 3

SQLite is a software library that provides a relational database management system. The lite in SQLite means lightweight in terms of setup, database administration, and required resources.

SQLite has the following noticeable features: self-contained, serverless, zero-configuration, transactional.

Serverless

Normally, an RDBMS such as MySQL, PostgreSQL, etc., requires a separate server process to operate. The applications that want to access the database server use TCP/IP protocol to send and receive requests. This is called client/server architecture.

The following diagram illustrates the RDBMS client/server architecture:

- RDBMS Client Server Architecture
- SQLite does NOT work this way.
- SQLite does NOT require a server to run.

SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write directly from the database files stored on disk.

The following diagram illustrates the SQLite server-less architecture:

What is SQLite?

Self-Contained

SQLite is self-contained means it requires minimal support from the operating system or external library. This makes SQLite usable in any environment especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc.

SQLite is developed using ANSI-C. The source code is available as a big sqlite3.c and its header file sqlite3.h. If you want to develop an application that uses SQLite, you just need to drop these files into your project and compile it with your code.

Zero-configuration

Because of the serverless architecture, you don’t need to “install” SQLite before using it. There is no server process that needs to be configured, started, and stopped.

In addition, SQLite does not use any configuration files.

Transactional

All transactions in SQLite are fully ACID-compliant. It means all queries and changes are Atomic, Consistent, Isolated, and Durable.

In other words, all changes within a transaction take place completely or not at all even when an unexpected situation like application crash, power failure, or operating system crash occurs.

SQLite distinctive features

SQLite uses dynamic types for tables. It means you can store any value in any column, regardless of the data type.

SQLite allows a single database connection to access multiple database files simultaneously. This brings many nice features like joining tables in different databases or copying data between databases in a single command.

SQLite is capable of creating in-memory databases that are very fast to work with.

Sqlite is Django Default Database

A database setup by itself can be time consuming. If you want the quickest setup to enable Django with a database leave the previous configuration as is. SQLite doesn't require additional credentials or Python packages to establish a Django database connection. Just be aware a SQLite database is a flat file and Django creates the SQLite database based on the NAME variable value. In the case of listing 1-14, under a Django project's BASE_DIR and as a flat file named db.sqlite3.

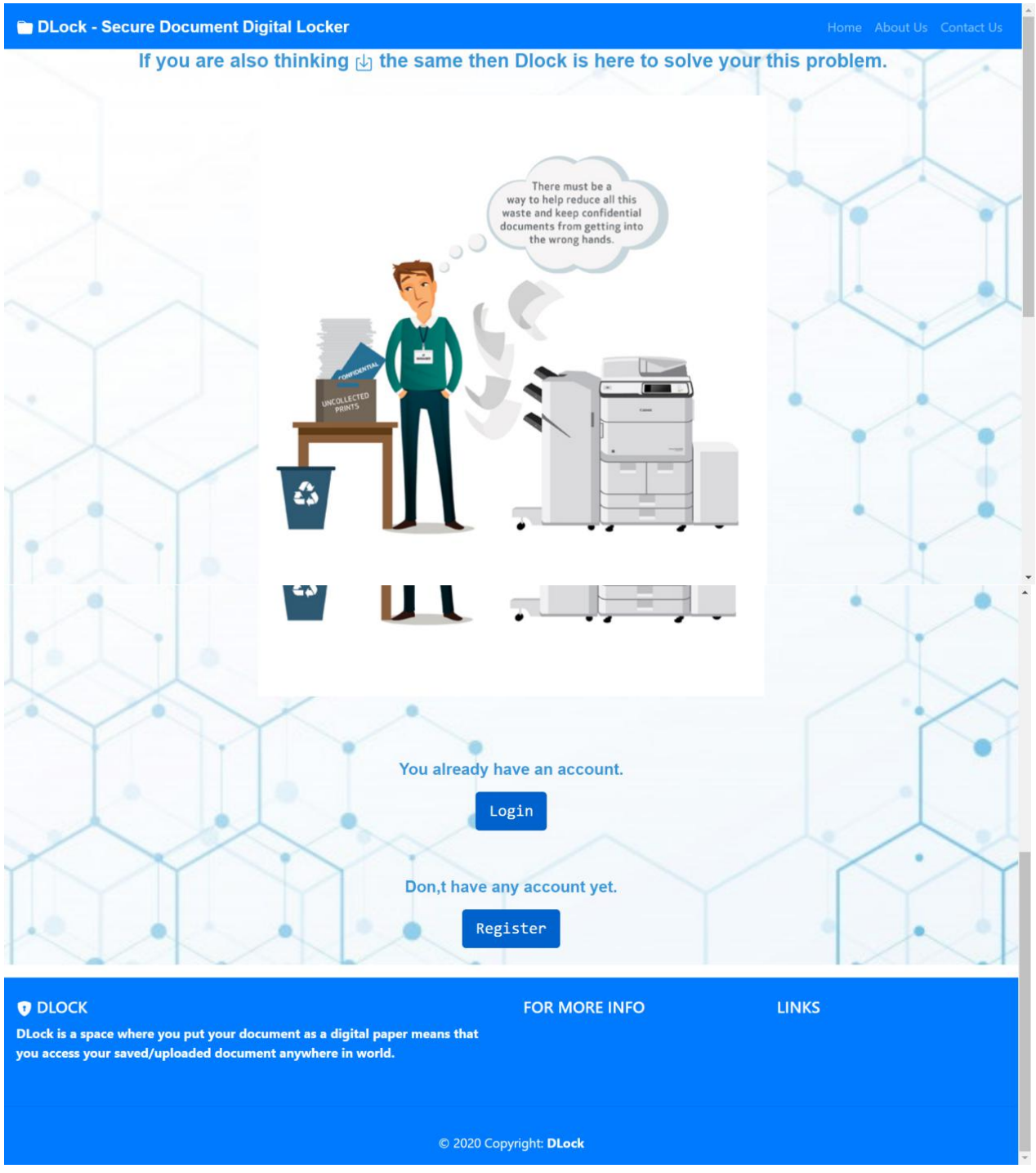
sqlite is the default for Django. It should not be used in production since it is usually slow.

#myapp/settings/settings.py

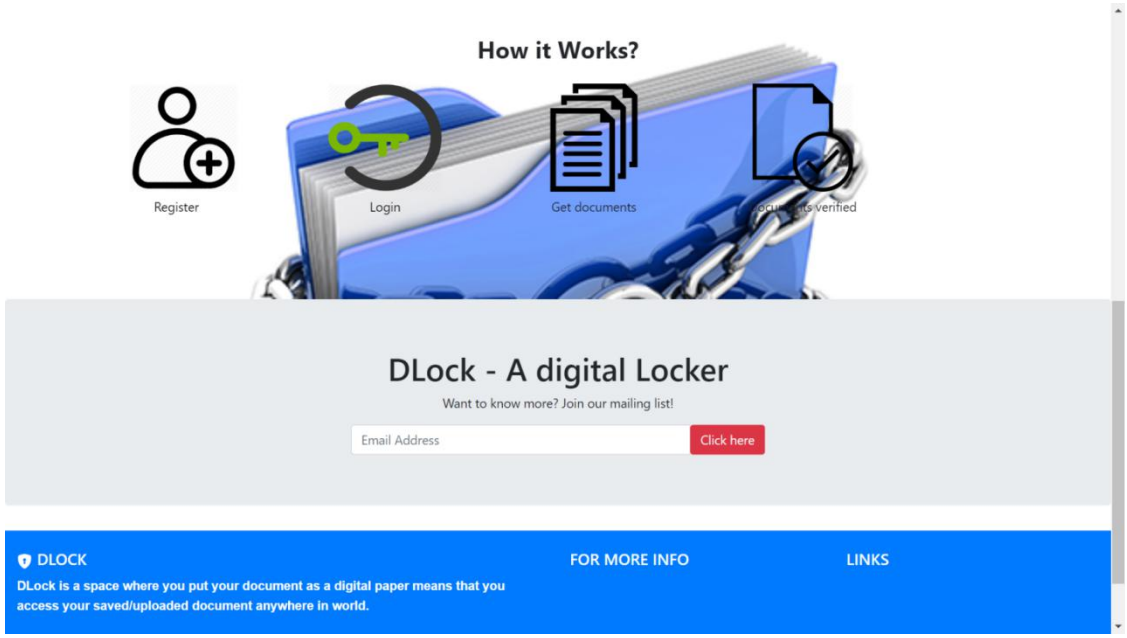
```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'db/development.sqlite3',
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    },
}
```

6. **OUTPUT SCREENS**

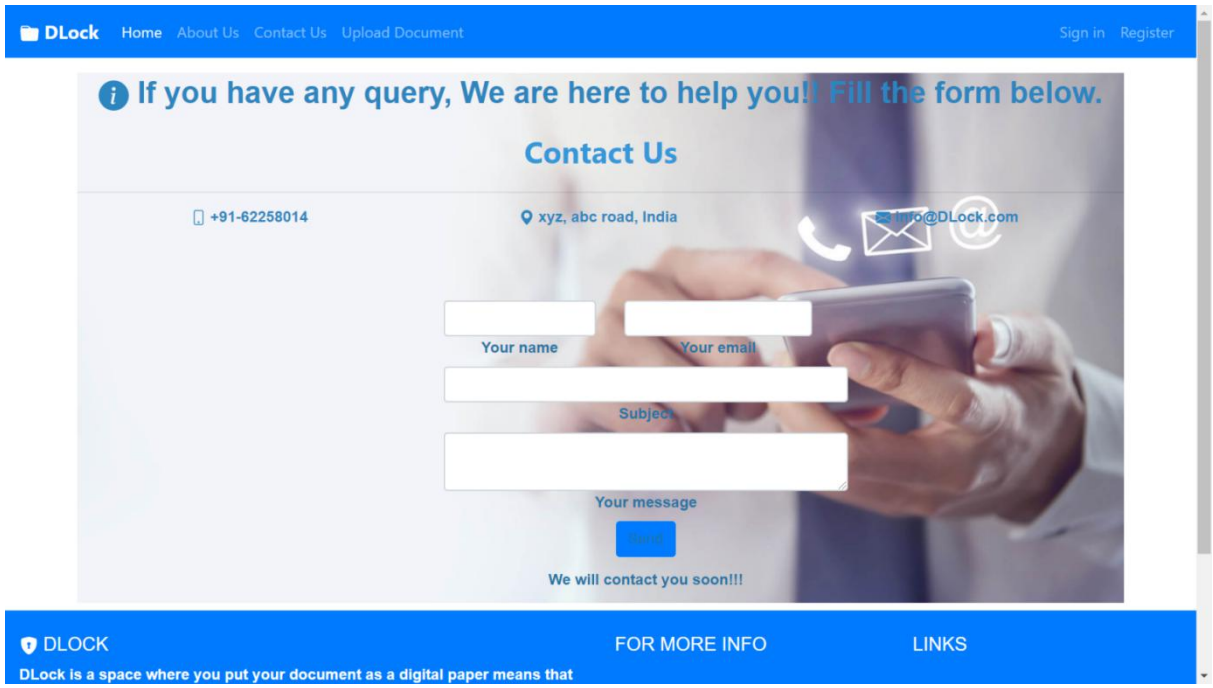
USER LOGIN


DLock - Secure Document Digital Locker
Home
About Us
Contact Us
If you are also thinking the same then Dlock is here to solve your this problem.
There must be a way to help reduce all this waste and keep confidential documents from getting into the wrong hands.
CONFIDENTIAL
UNCOLLECTED PRINTS
You already have an account.
Login
Don,t have any account yet.
Register
DLOCK
DLock is a space where you put your document as a digital paper means that you access your saved/uploaded document anywhere in world.
FOR MORE INFO
LINKS
© 2020 Copyright: DLock

USER HOME PAGE

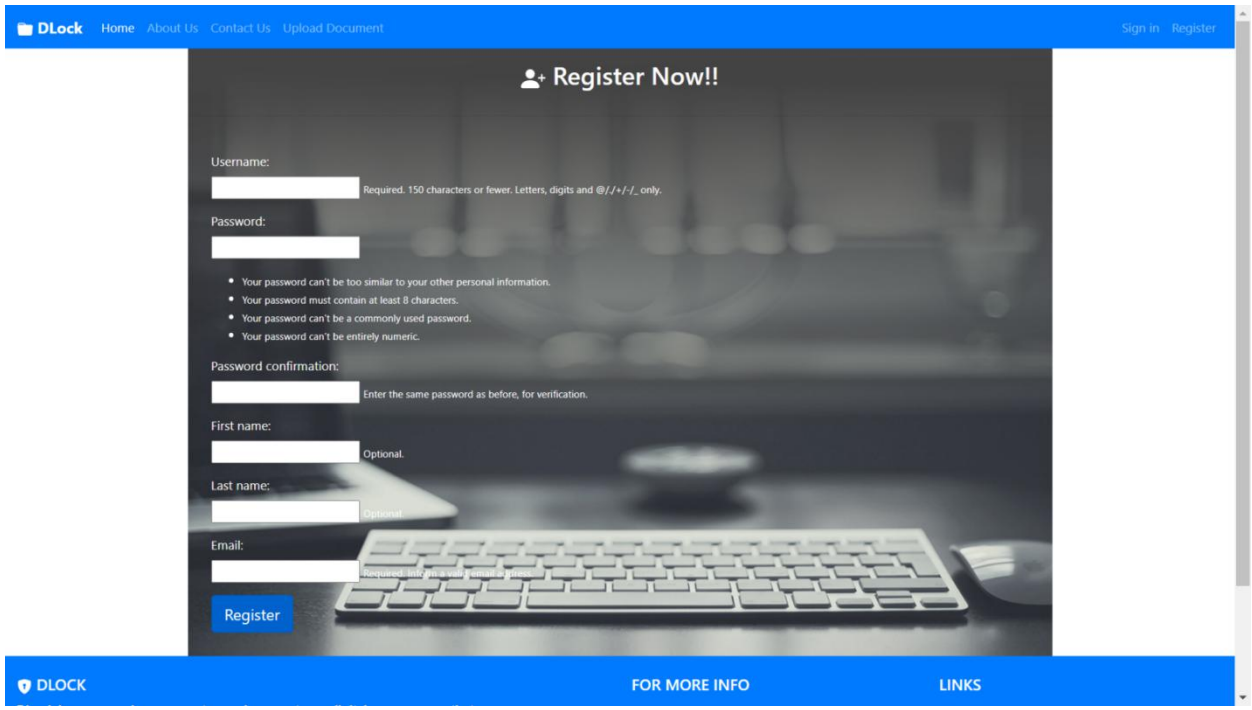


CONTACT US PAGE

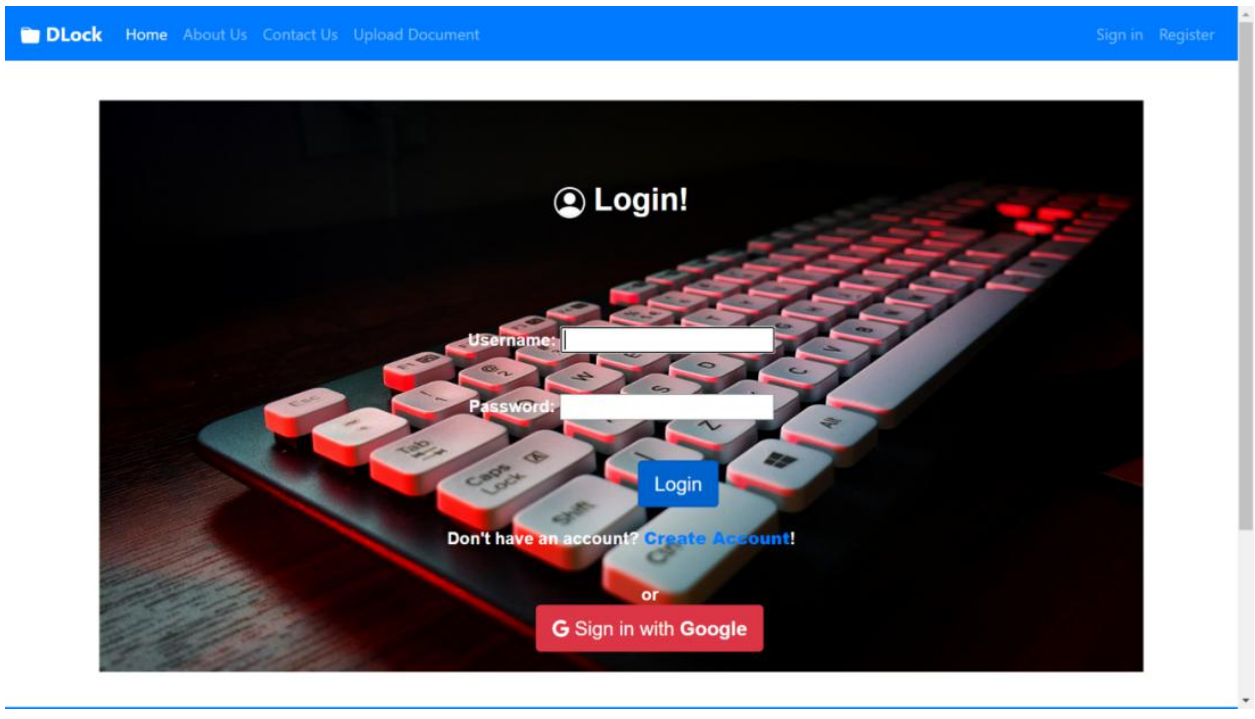


ABOUT US PAGE

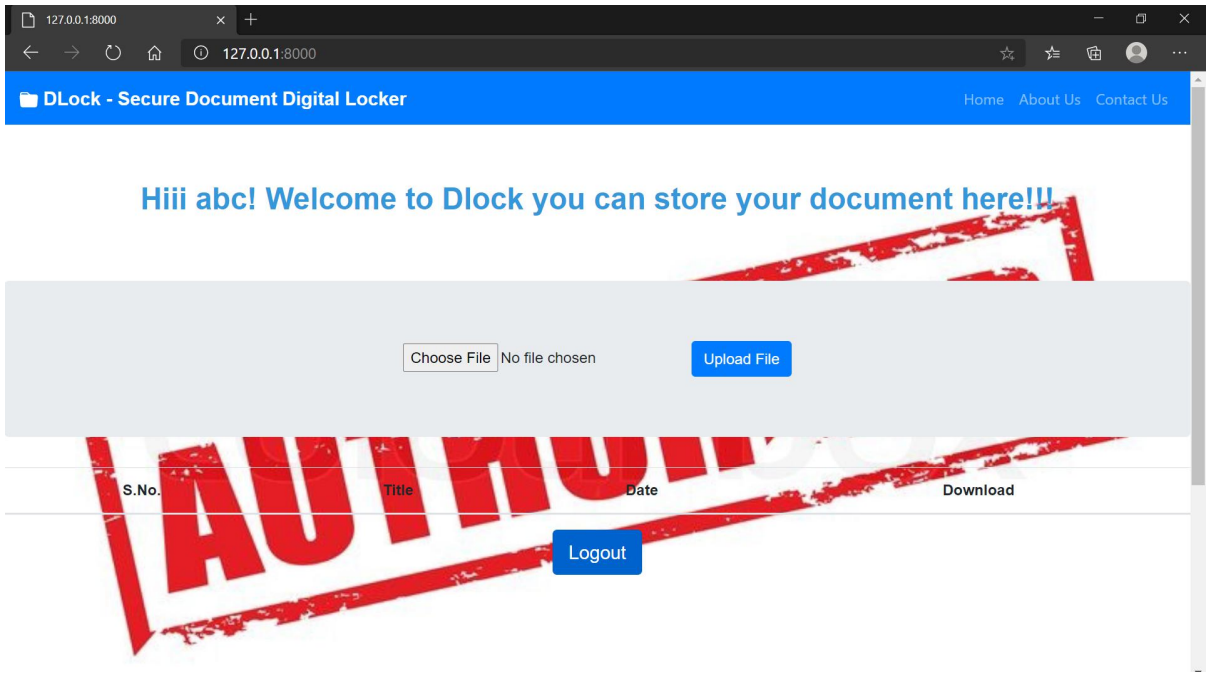
USER REGISTRATION PAGE



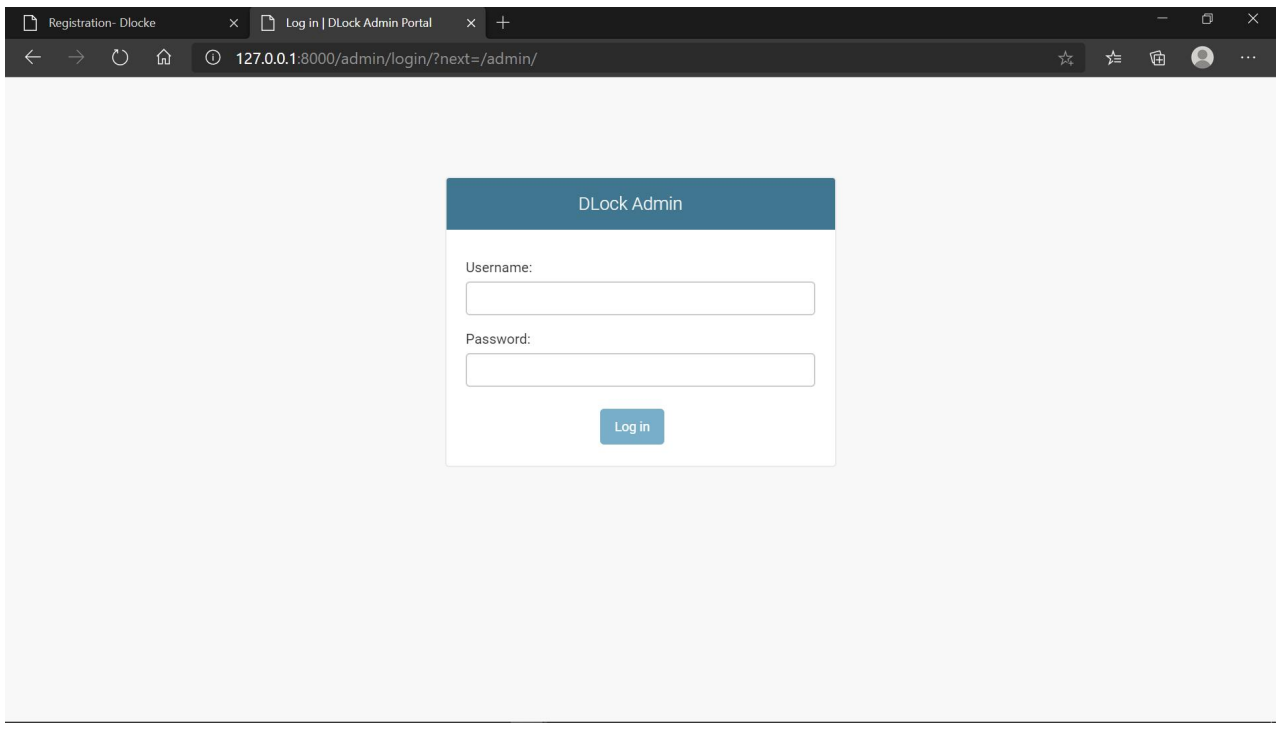
LOGIN PAGE



AFTER LOGIN PAGE



ADMIN LOGIN PAGE



7. SYSTEM SECURITY

7.1 INTRODUCTION

The protection of computer based resources that include hardware, software, data, procedures and people against unauthorized use or natural Disaster is known as

System Security.

System Security can be divided into four related issues:

- Security
- Integrity
- Privacy
- Confidentiality

SYSTEM SECURITY refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

DATA SECURITY is the protection of data from loss, disclosure, modification and destruction.

SYSTEM INTEGRITY refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

PRIVACY defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

CONFIDENTIALITY is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

7.2 SECURITY SOFTWARE

It is the technique used for the purpose of converting communication. It transfers message secretly by embedding it into a cover medium with the use of information hiding techniques. It is one of the conventional techniques capable of hiding large secret message in a cover image without introducing many perceptible distortions.

NET has two kinds of security:

- Role Based Security
- Code Access Security

The Common Language Runtime (CLR) allows code to perform only those Operations that the code has permission to perform. So CAS is the CLR's security system that enforces security policies by preventing unauthorized access to protected **resources** and **operations**. Using the Code Access Security, you can do the following:

- Restrict what your code can do
- Restrict which code can call your code
- Identify code

8. CONCLUSION

8.1 CONCLUSION

DLock - Digital Locker scheme is a good initiative taken ensure safe custody of the important documents such as PAN card; AdhaarCard , DL etc. which determines the nationality of the Citizens of India electronically. This move would in turn result in much more transparency, authenticity and eradication of red tapism and corruption to the maximum extent possible. As DLock improves it will be capable to provides access to 2,39,83,48,579 authentic digital documents (PAN Verification Record, Driving License, Vehicle RC, Aadhaar, Educational and Other Certificates). This is a good move towards making India ‘a digitized economy’ and thereby ensuring broadband connectivity in the rural areas. Thus, for the program to be a success, it becomes imperative that there is absolute coordination between the various departments and utmost commitment on the part of government to make sure that each and every aspect of this programme is pushed to show the results. But it is necessary to overcome all the challenges mentioned here with the help of proper IT security, IT literacy to successfully materialize the dream of a “Digital India” and also to implement and execute this new-age idea across different agencies/govt departments/institutions. These legacy documents can be electronically stored in our webapp. This project can be used by the students to store their private documents safely with more security. Its main aim is to eliminate carrying of documents physically. Each student will have their own username and password to login to their respective accounts. Since admin issues documents directly to the students according to their USN, it overcomes fraudness and also it reduces the time spent by the students to collect hardcopy of the documents from the administrative office. It also makes it easy for students to receive services.

COMPANY PROFILE

About Techlene Software Solutions

Techlene is a software solution, ERP products, and Big Data services company headquartered in Indore, INDIA. Techlene is focused on creating new ways of ERP & analyzing big data for businesses & Educational Institution - helping them gain key business insights across the enterprise. We bring a unique mix of Data Science expertise across the Big Data ecosystem & ERP including Hadoop, NoSQL, PIG, HIVE, machine learning, and visualization.

We integrate accounting, payroll, and manufacturing, sales & inventory systems, supporting the ambition of entrepreneurs and business builders. Today, business builders measure success in strong relationships, partnerships, and communities. It's why Techlene helps drive today's business builders with the most intelligent and flexible Smart ERP Solution and advice to manage everything from money to people.

Our Product Range:

- Manufacturing ERP Software
- Trading ERP Software
- Industrial ERP Software
- HR and Payroll Management Software
- Hospital Management System
- School Management System
- College Management System
- PoS Software
- Retail Software

We have successfully deployed our ERP solutions in following Industries and Verticals:

- Electronics Manufacturing Industry
- Furniture Manufacturing Industry
- Plastics Manufacturing Industry
- Steel Manufacturing and Trading Industry
- Construction Industry
- Medical and Healthcare Organizations
- Educational Institutes including School, College, and Training Institutes

Techlene is a pioneer in providing IT Solutions and Services including:

- Custom Software Development
- Custom Business Application Software
- Custom Mobile Application Development
- Custom ERP Software
- Big Data Solutions
- Data Analytics Solutions
- Cloud Software Solutions
- Blockchain Application Development
- Android App Development
- iPhone App Development
- iPad App Development

We provide innovative Solutions

We are an adept team of Software Analysts, Programmers, Graphic designers, Software testers and technology enthusiasts who share a common goal of bringing success to our clients. We believe in performance and hence we deliver result-driven products.

Techlene is a fast growing enterprise software player disrupting the market with its multi-tenanted mobile-based & cloud enterprise software in the area of ERP, HR-Payroll & CRM. Techlene Software Solutions focuses on Innovation and Smart Automation to differentiate itself in the marketplace. On the Innovation front, Techlene has been focusing on moving towards Smart Industries Solution and Smart Campus Solution.

Specialties

Custom ERP Software Development,

Big Data Analysis, Hadoop, Business Anlaysis,

Manufacturing ERP Software,

Industrial ERP Software,

Retail Analysis Software,

School Management System,

College Management System,

Payroll Management Software,

CRM Software, Android Development,

Big Data Solutions,

Big Data Services,

Java Development,

Software Development,

Python,

Custom Business Application Development,

Custom Mobile App Development,

Custom Mobile Application Development,

ERP Software for Electronics Company,

ERP Software for Furniture Company,

ERP Software for Plastics Company,

Construction ERP Software,

Hospital Management System,

and HRMS Software.

For more details about **Techlene Software Solutions,** please visit

India :- P101, Metro Tower, Near Vijaynagar Square,Indore(MP), INDIA, Pin - 452010

Australia :- 46, Chapel Street Windsor VIC - 3181, Australia

Official website :- https://www.techlene.com/

BIBLIOGRAPHY

- ✓ DigiLocker - Online document storage facility, National Portal of India https://digilocker.gov.in/
- ✓ https://indianexpress.com/article/india/india-others/centres-digital-locker-to-empower-citizens/#sthash.dPsccr2K.dpuf
- ✓ A Wedge of Django: Covers Python 3.8 and Django 3.x - Daniel and Audrey Feldroy
- ✓ Speed Up Your Django Tests - Adam Johnson
- ✓ Django for Professionals: Production websites with Python & Django - William S Vincent
- ✓ Django for Beginners: Build websites with Python and Django - William S Vincent
- ✓ Django Design Patterns and Best Practices - Second Edition: Industry-standard web development techniques and solutions using Python - Arun Ravindran
- ✓ Lightweight Django - Julia Elman
- ✓ Django for APIs: Build web APIs with Python and Django - William S Vincent
- ✓ Django: Web Development with Python - by Samuel Dauzon, Aidas Bendoraitis, Arun Ravindran