## CHAPTER 1

**SYNOPOSIS**

Course and certificate management (C&CM) provides a simple interface for maintenance of student information by showing certificate. It can be used by educational institutes or colleges to maintain the records of students easily as well as it helpful for company in terms of showing certificate. The creation and management of accurate, up-to-date information regarding a students’ academic career is critically important in the university as well as colleges. Course and certificate management deals with all kind of student details, academic related reports, college details, course details, curriculum, batch details, and other resource related details too. It tracks all the details of a student from the day one to the end of the course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters, years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result and all these will be available through a secure, online interface embedded in the college’s website. It will also have faculty details, batch execution details, students’ details in all aspects, the various academic notifications to the staff and students updated by the college administration. It also facilitate us explore all the activities happening in the college, Different reports and Queries can be generated based on vast options related to students, batch, course, faculty, exams, semesters, certification and even for the entire college.

In this Course and Certificate Management ,we are using various languages, various frameworks and databases these are the requirements for our project, here we use front end which is python all the entire code and for running in command prompt and in addition of that we are also using bootstrap which is used for free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components. XAMPP stands for Cross-Platform (X), Apache (A), (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database , and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Django is a high-level **,** Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Coming to backend we used Mysql it is used for storing the data and to retrive back the data that is what we entered. MySQL is a free, open-source database management system (DBMS for short). A DBMS is a system that manages databases and connects them to software. For example, a MySQL database can be used to run a website, to run the database of an ERP or any other software.  
  
MySQL is a powerful, free open-source database management system that has been around for years. It is very stable and has a big community that helps maintain.  
  
MySQL might not be as popular for larger systems that will mostly run on Microsoft SQL Server or Oracle. These proprietary DBMS are more scalable, have more resources available on the market and have more advanced features that MySQL.

By using all this requirements we are ready to do a Course and Certificate Management.

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 PROBLEM DEFINITION**

Literature [survey](http://www.blurtit.com/q876299.html) is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the [programmers](http://www.blurtit.com/q876299.html) start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from [book](http://www.blurtit.com/q876299.html) or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

**2.2 PROJECT DETAILS**

Although many randomized programming languages have been designed throughout the years , only recently one implementation of a languages or frame works were updated to day by day of randomized multicast and agreement frameworks has been reported on this days ,we are ready to doing a Course and Certificate Management for each tool or language we are used a latest technology which is not form a any risk to this project. In this we used a python, mysql , Django framework, sublime text 3 which is a text editor for any programming , here we also used. According to this project we want to do , it means we want to install all this tools and languages which are various versions.

All the coding part of the python and bootstrap, cascade style sheet, hyper text markup language we are used in the sub line text3 it takes or fetched inside the code and we also take a help from browser to execute the project by initialize in the uniform resource locator by internet protocal by hosting that files.

**CHAPTER 3**

**COMPUTATIONAL ENVIRONMENT**

**3.1 SOFTWARE SPECIFICATIONS**

* Operating system : Windows XP Professional, windows 7/8/9/10
* Front End : Python 3.2.6,Django Frame work, sublime text 3,Bootstrap,Xampp web server.
* Back End : Mysql

**3.2 HARDWARE SPECIFICATIONS**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 256 Mb.

**3.3 SOFTWARE FEATURES**

## 3.3.1 PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language.

Python is designed to be highly readable. It uses English keywords frequently whereas the

other languages use punctuations. 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 1.0 was released in November 1994. In 2000, Python 2.0 was released.

Python 2.7.11 is the latest edition of Python 2.

Meanwhile, Python 3.0 was released in 2008. Python 3 is not backward compatible

with Python 2. The emphasis in Python 3 had been on the removal of duplicate

programming constructs and modules so that "There should be one -- and

preferably only one -- obvious way to do it." Python 3.5.1 is the latest version of

python 3.

**Python Features**

Python's features include-

**Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined

syntax. This allows a 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. A 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.

o add the Python directory to the path for a particular session in Windows-

**At the command prompt :** type

path %path%;C:\Python and press Enter.

**Note:** C:\Python is the path of the Python directory.

# DJANGO

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django makes it easier to build better web apps quickly and with less code.

**Note**: Django is a registered trademark of the Django Software Foundation, and is licensed under BSD License.

**History of Django**

**2003**: Started by Adrian Holovaty and Simon Willison as an internal project at the Lawrence Journal-World newspaper.

**2005**: Released July 2005 and named it Django, after the jazz guitarist Django Reinhardt.

**2005**: Mature enough to handle several high-traffic sites.

**Current**: Django is now an open source project with contributors across the world.

**Django – Design Philosophies**

Django comes with the following design philosophies:

**Loosely Coupled**: Django aims to make each element of its stack independent of the others.

**Less Coding**: Less code so in turn a quick development.

**Don't Repeat Yourself (DRY)**: Everything should be developed only in exactly one place instead of repeating it again and again.

**Fast Development**: Django's philosophy is to do all it can to facilitate hyper-fast development.

**Clean Design**: Django strictly maintains a clean design throughout its own code and makes it easy to follow best web-development practices.

**Advantages of Django**

Here are few advantages of using Django which can be listed out here: Django

6

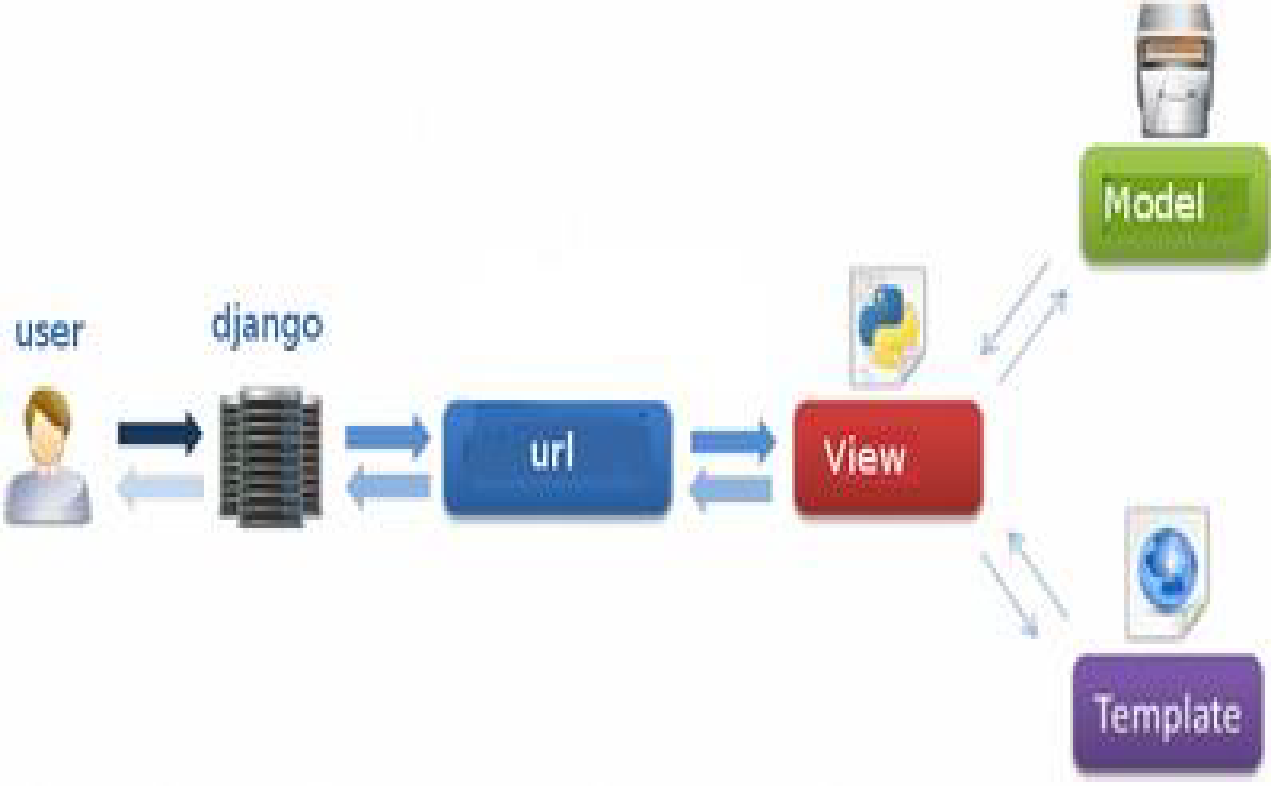
**Object-Relational Mapping (ORM) Support**: Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django-nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.

**Multilingual Support**: Django supports multilingual websites through its built-in internationalization system. So you can develop your website, which would support multiple languages.

**Framework Support**: Django has built-in support for Ajax, RSS, Caching and various other frameworks.

**Administration GUI**: Django provides a nice ready-to-use user interface for administrative activities.

**Development Environment**: Django comes with a lightweight web server to facilitate end-to-end application development and testing. Django



**MVC Pattern**

When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller. Check our MVC tutorial here to know more.

**DJANGO MVC - MVT Pattern**

The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).

The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request:

The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user. Django

ENVIRONMENT

Django development environment consists of installing and setting up Python, Django, and a Database System. Since Django deals with web application, it's worth mentioning that you would need a web server setup as well.

**Step 1 – Installing Python**

Django is written in 100% pure Python code, so you'll need to install Python on your system. Latest Django version requires Python 2.6.5 or higher for the 2.6.x branch or higher than 2.7.3 for the 2.7.x branch.

If you're on one of the latest Linux or Mac OS X distribution, you probably already have Python installed. You can verify it by typing *python* command at a command prompt. If you see something like this, then Python is installed.

$ python

Python 2.7.5 (default, Jun 17 2014, 18:11:42)

[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2

Otherwise, you can download and install the latest version of Python from the link http://www.python.org/download.

**Step 2 - Installing Django**

Installing Django is very easy, but the steps required for its installation depends on your operating system. Since Python is a platform-independent language, Django has one package that works everywhere regardless of your operating system.

You can download the latest version of Django from the link http://www.djangoproject.com/download.

**UNIX/Linux and Mac OS X Installation**

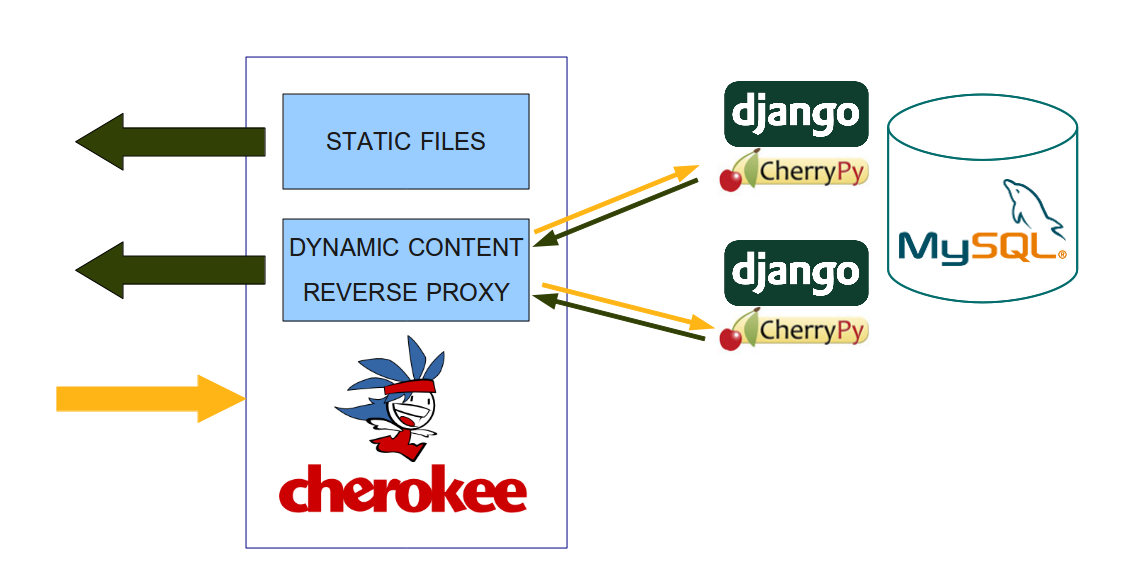
You have two ways of installing Django if you are running Linux or Mac OS system:

You can use the package manager of your OS, or use easy\_install or pip if installed.

Install it manually using the official archive you downloaded before.

We will cover the second option as the first one depends on your OS distribution. If you have decided to follow the first option, just be careful about the version of Django you are installing. Django

9



Let's say you got your archive from the link above, it should be something like Django-x.xx.tar.gz:

Extract and install.

$ tar xzvf Django-x.xx.tar.gz

$ cd Django-x.xx

$ sudo python setup.py install

You can test your installation by running this command:

$ django-admin.py --version

If you see the current version of Django printed on the screen, then everything is set.

**Note**: For some version of Django it will be django-admin the ".py" is removed.

**Windows Installation**

We assume you have your Django archive and python installed on your computer.

First, PATH verification.

On some version of windows (windows 7) you might need to make sure the Path system variable has the path the following C:\Python27\;C:\Python27\Lib\site-packages\django\bin\ in it, of course depending on your Python version.

Then, extract and install Django.

c:\>cd c:\Django-x.xx

Next, install Django by running the following command for which you will need administrative privileges in windows shell "cmd":

c:\Django-x.xx>python setup.py install

To test your installation, open a command prompt and type the following command:

c:\>django-admin.py --version

If you see the current version of Django printed on screen, then everything is set.

OR

Launch a "cmd" prompt and type python then: Django

10

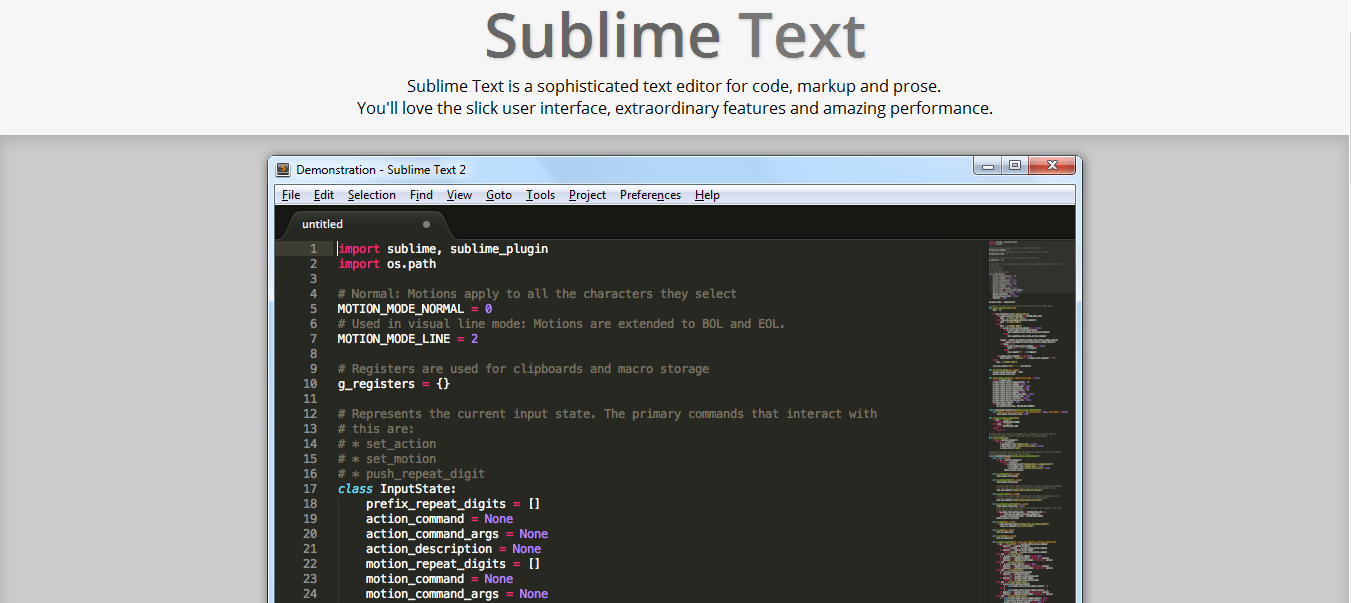
c:\> python

>>> import django

### >>> print django.get\_version()

**3.3.3 SUBLIME TEXT 3**

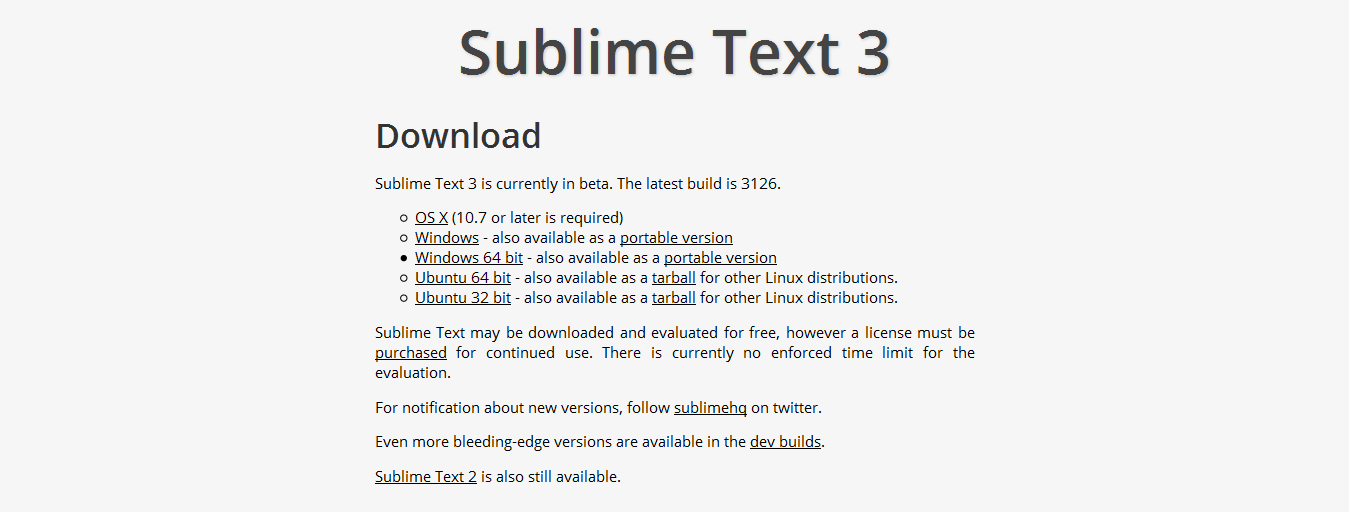
How to Install Sublime Text 3 in Windows:



In order to write and edit your PHP code you need a text editor. [Sublime Text](https://www.sublimetext.com/) is a light weight text editor but has a bunch of features just close to an IDE. Besides, it’s free to download and install on your computer.

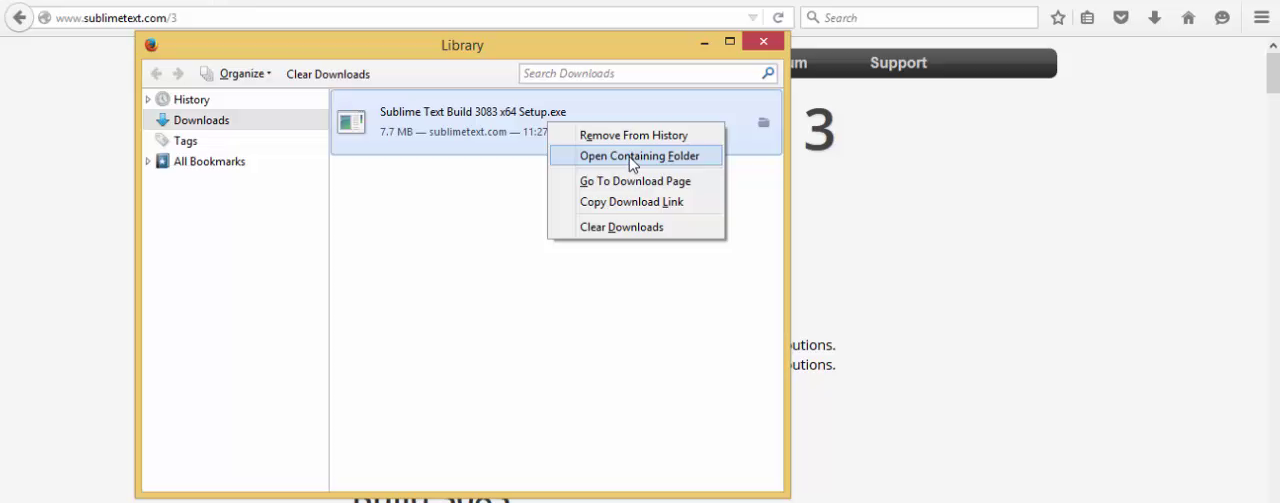
You can watch this tutorial on YouTube on[How to Install Sublime Text 3](https://www.youtube.com/watch?v=49ZcIc0ob7c&list=PLOR_GQqc11FAgKXQOXtGX-Z_o6WwlMrJ9&index=3) in Windows using this link.

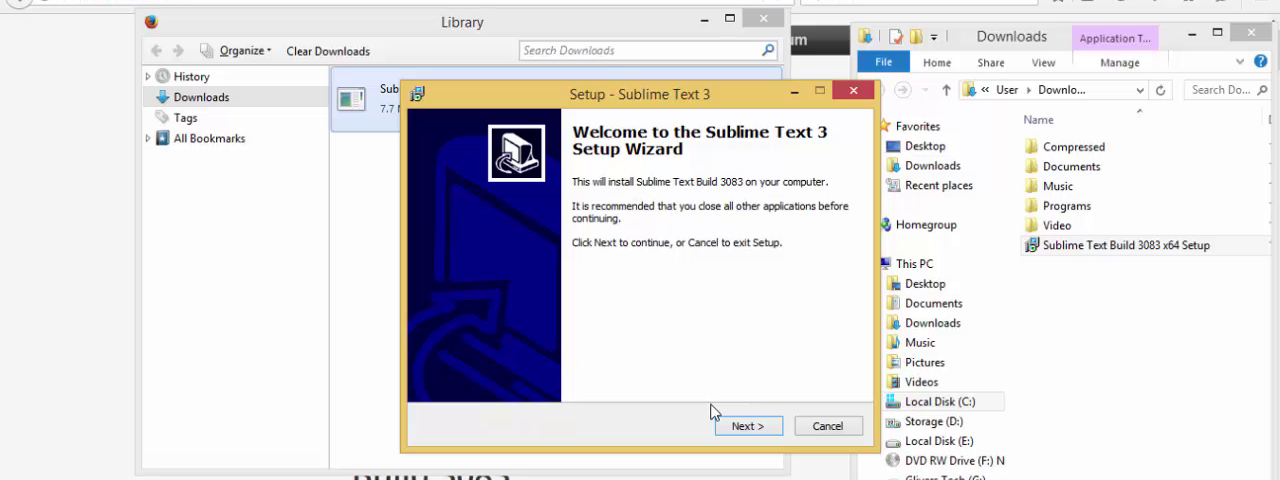
To download and install Sublime Text first you need to download the Sublime Text installer for Windows from the [Sublime Text 3](http://sublimetext.com/3) official website.

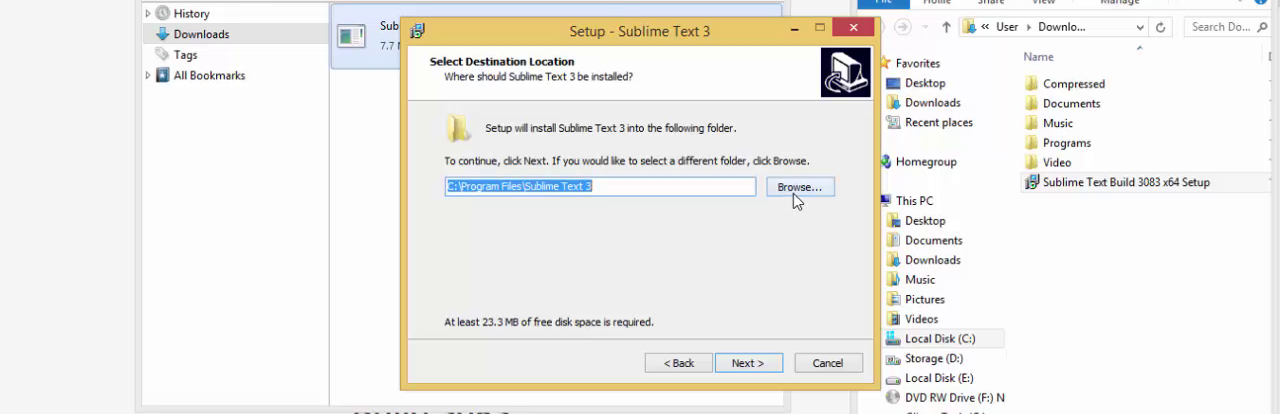
Since we are interested in installing Sublime Text on Windows Operating System, we will chose the link to download the setup for Windows. If you are running Windows 64-bit then you will need to download the 64-bit installer, otherwise you will just download the Windows(32-bit) Installer.

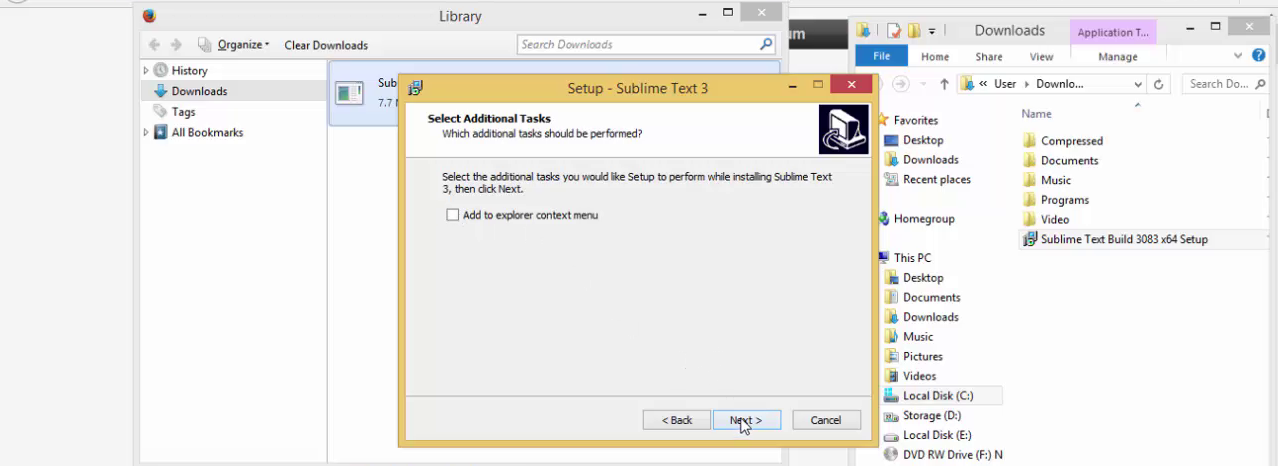
Once you click on the download link, the dialog below will appear. Click on Save File to download and save the installer to your local drive.

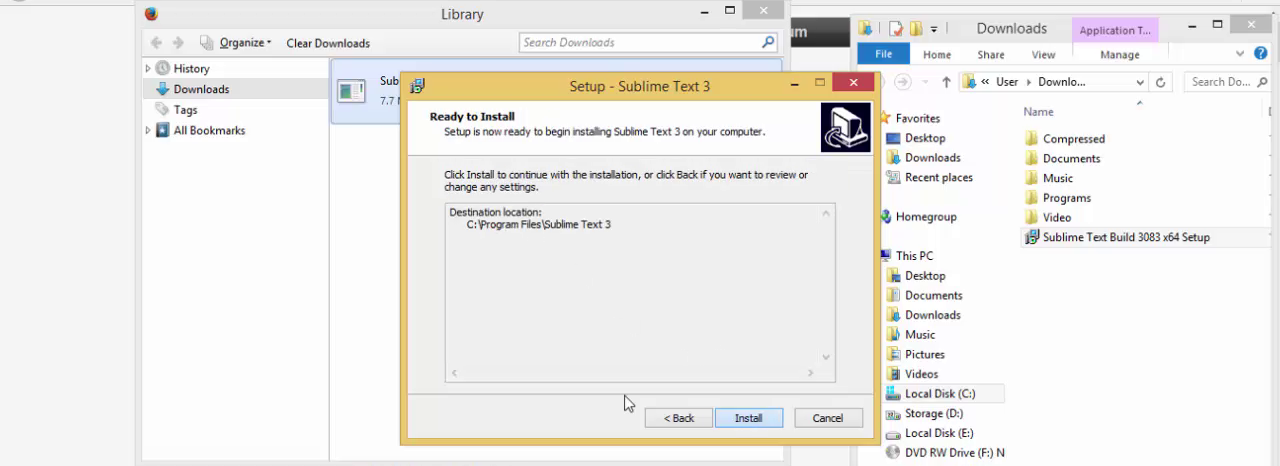
Once the download is complete, right click on the file name in the browser Downloads window and then click on Open Containing Folder to open the folder where the installer was saved.

Double click the setup file in order to launch the Sublime Text Installer as shown below.

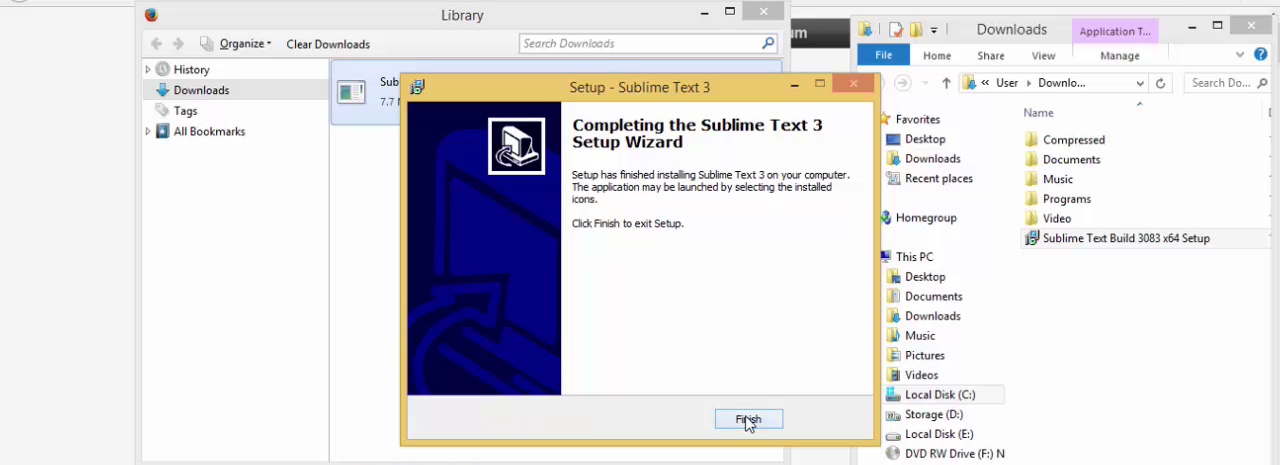
Click on Next to move the the next step of the installation. In the next step you can choose the directory where you want sublime text to be installed. The default directory for sublime text is Local Drive C. However, if you desire to have sublime text installed in a different directory you can specify the file path in the text entry field or click on the file browser to select a folder.

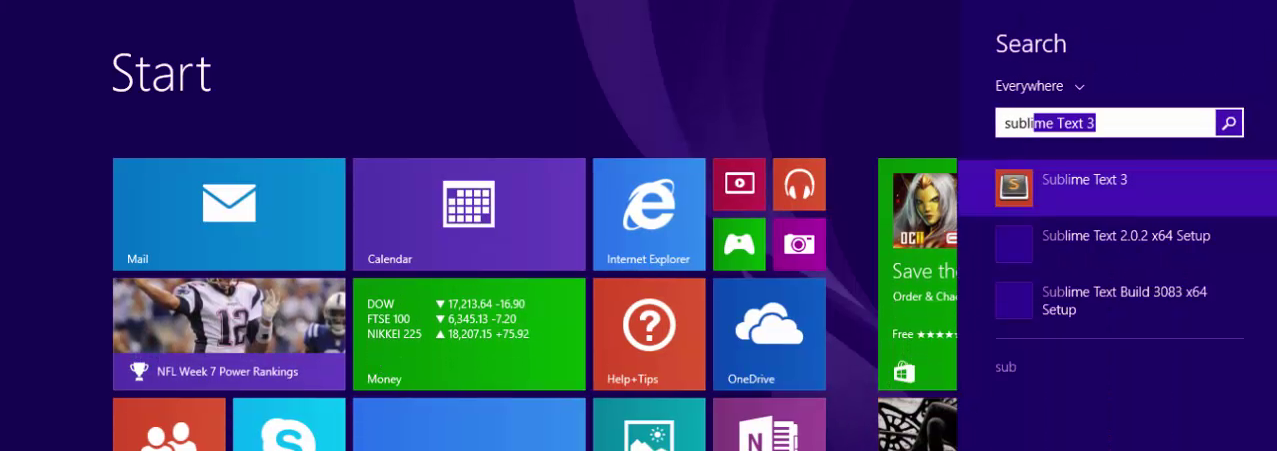
After specifying the installation directory click on the Next button in order to advance to the next step of the installation.

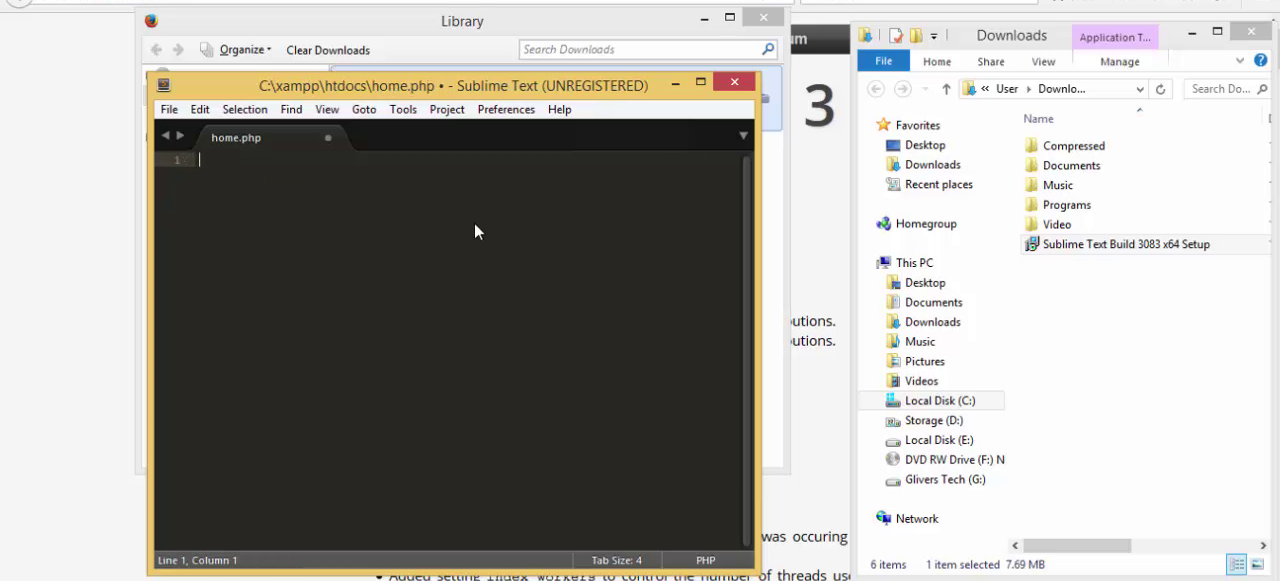
Leave the checkbox unchecked and click Next to move to the next step of the installation process for sublime text.

Sublime Text in now ready to install. Click on Install to start installing Sublime Text 3 locally on your computer.

Depending on the processing power and memory of your computer, sublime text might take a few seconds to a few minutes to install. Once the installation is complete, click on the Finish button as shown below to close the installer window.

Sublime Text should now have been installed in your computer. In order to launch and use your sublime text, Tap the Windows key and then use the Windows Search bar to search for the launcher. Type sublime text in the text input field as shown below. Once the Sublime Text 3 launcher appears, click on the name to launch sublime text.

This would launch the sublime text window as shown below.

Sublime Text is now successfully installed on your Windows computer. You can now start coding your dream project!

If you followed the steps outlined above you should be having a working installation of Sublime Text.

.

**3.3.5 XAMPP**

XAMPP stands for Cross-Platform(X ) , Apache (A) , MySQL (M) , PHP (P) and Perl (P) { Practical Extraction and Report Language}.

2.It is a simple , light weight Apache distribution that makes it extremely easy for developers to create a local webserver for testing purposes.

3.Everything you need to set up a webserver–server application (Apache),database (MySQL) ,and scripting language (PHP)–is included in as imple extractable file.

4.XAMPPisalsocross-platform,whichmeansitworksequallywellonLinux,MacandWindows.SincemostactualwebserverdeploymentsusethesamecomponentsasXAMPP,itmakestransitioningfromalocaltestservertoaliveserverisextremelyeasyaswell

**XAMPPhasfourprimarycomponents.Theseare:**

**Apache:**Apache is the actual web server application that processes and delivers webcontent to a computer .Apache is the most popular webserver online ,powering nearly 54% of all websites.

**MySQL:**Every web application , how so evers imple or complicated , requires a database for storing collected data .MySQL ,which is open source ,is the world’s most popular database management system.

**PHP:**PHP stands for Hypertext Preprocessor.It is a server-side scripting language that powers some of the most popular websites in the world ,including Word Press and Facebook.

**Perl**:Perlisahigh-level,dynamic programming language us d extensively in network programming ,system admin,etc.Although less popular for web development purposes ,Perl has a lot of applications.

**Installing XAMPP**

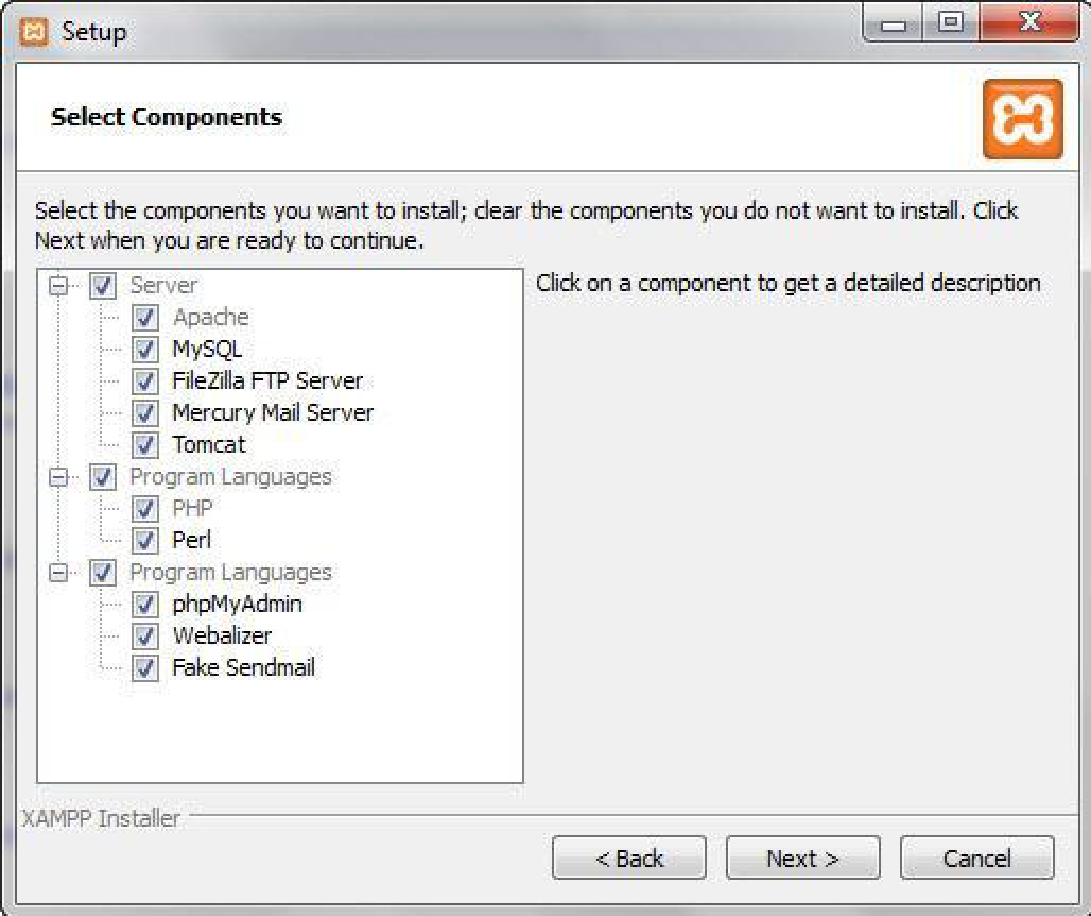
Follow these steps for installing XAMPP:

# Step1:Disableyouranti-virus as it an causes come XAMPP components to be have erratically.

**Step2**:Disable User Account Control (UAC). UAC limits write permission to XAMPP’s default installation directory (c:/ProgramFiles/xampp) ,forcing you to install in separate directory .You can learn how to disable UAChere .(Optional)

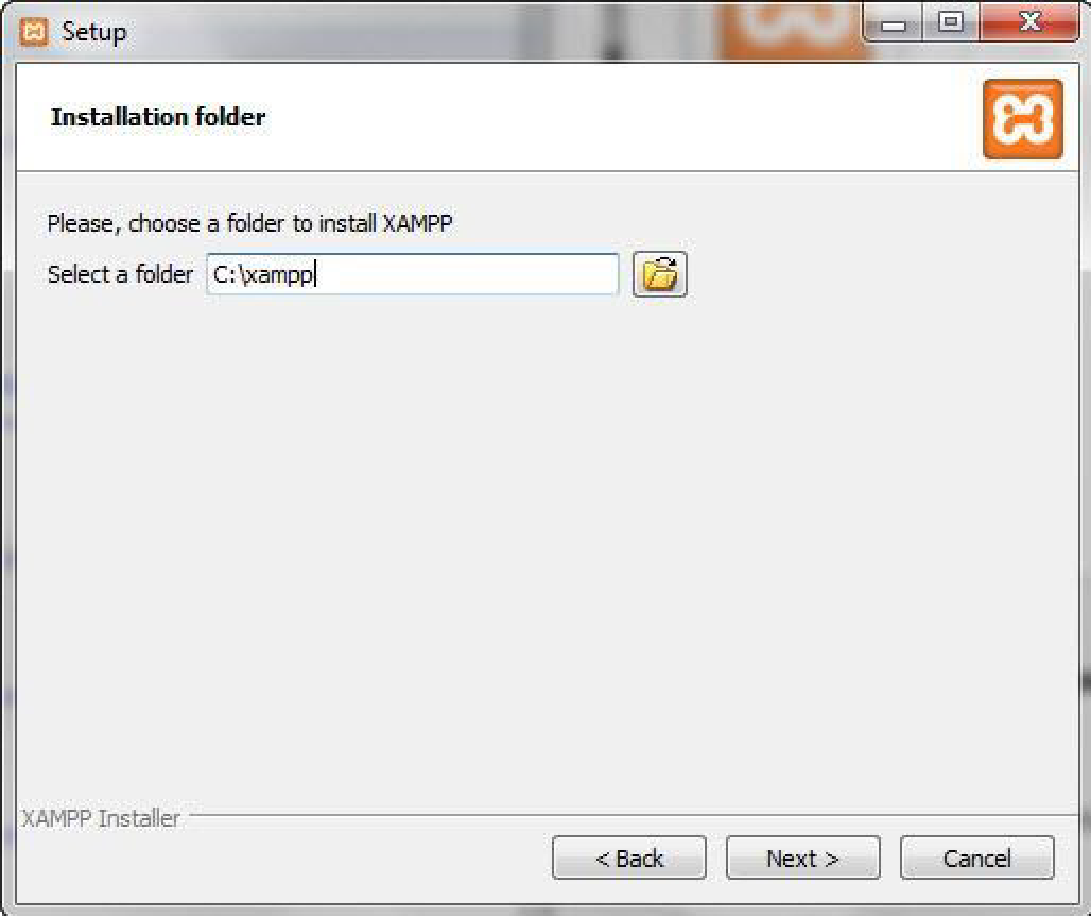
**Step3:**Start the installation process by double-clicking on the XAMPP

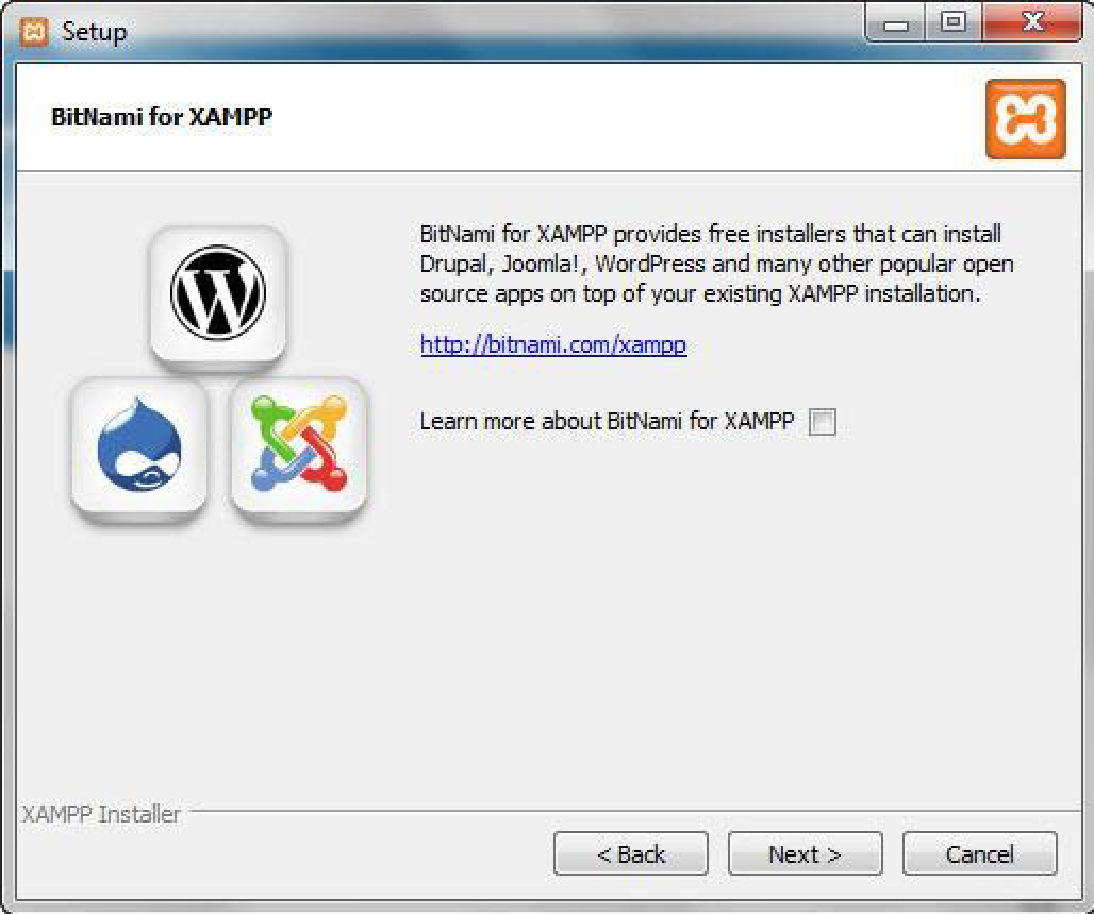
installer .Click‘ Next’ after the splash screen.

**Step4**:Here,you can select the components you want to install ll.Choose the default selection and click ‘Next’.

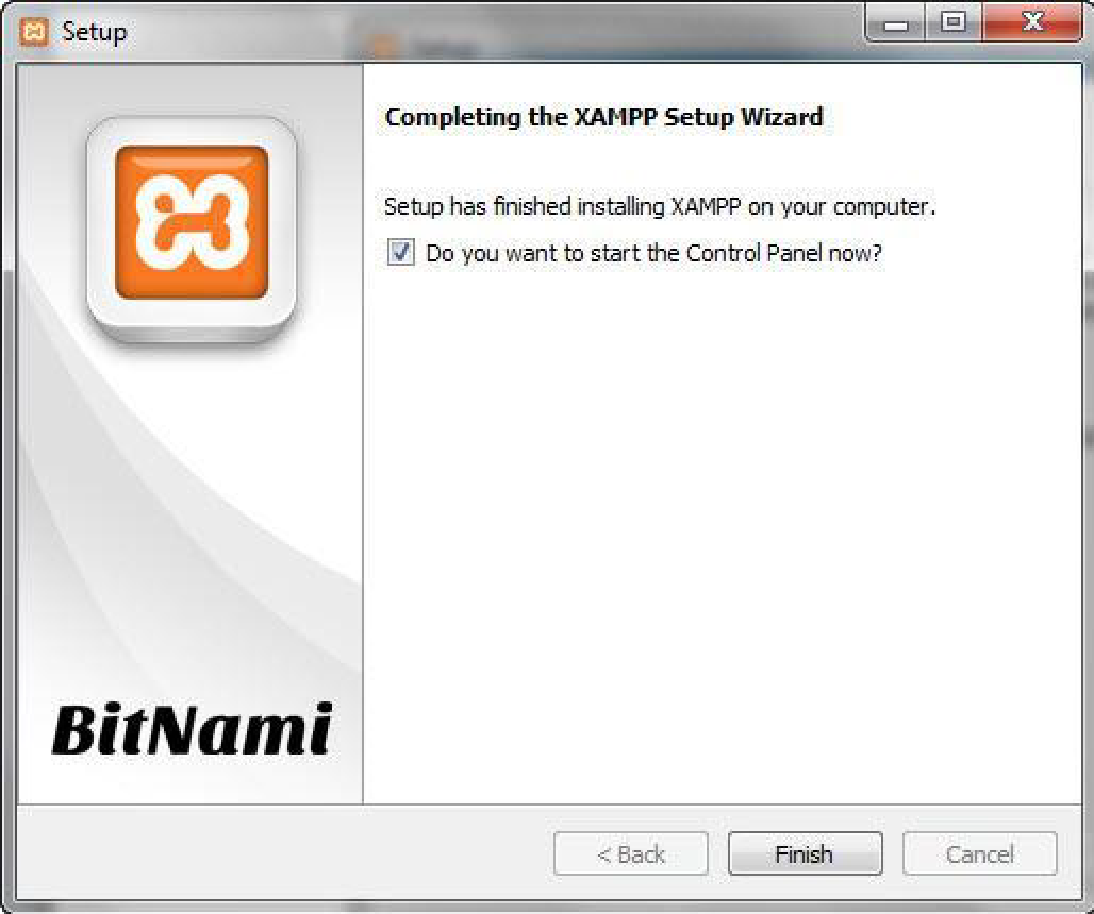
so make sure to select a drive thath as plenty of space

**Step5**:Choose the folder you want to install XAMPP in .This folder will hold all your web application files,

. 

**Step6:**Then exit screen is a promo for BitNami ,an appstore for server software .Deselect the ‘Learn more about BitNami for XAMPP’ checkbox, unless you actually enjoy receiving promo mails! 

**Step7:**Setup is now ready to install XAMPP . Click Next and wait for the installer to unpack and install selected components. This may take a few minutes. You may be asked to approve Firewall access to certain components ( such as Apac

**Step8:**Installation is now complete! Select the ‘Do you want to start the Control Panel now ?’check box to open the XAMPP control panel.

**Understanding XAMPP Control Panel**

The XAMPP control panel gives you complete control overall installed

XAMPP components .You can use the CP to start / stop different modules ,launch the Unix shell , open Windows explorer and see all operations running in the background.

HereisaquickoverviewoftheControlPanel.Fornow,youonlyneedtoknowhowtostartandstopanApacheserver.

**CHAPTER 4**

**SYSTEM PLANNING**

**4.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**4.1.1 Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### 4.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**4.1.3 Social Feasibility**

The aspect of study is to get the certificate through this system by the admin. This includes the process of training the students as well as employees to use the reduce the maintenance. The customer must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the admin solely depends on the students or employers to train their courses about the project given by their companies and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as to get a certificate through online.

**CHAPTER 5**

**SYSTEM ANALYSIS**

**5.1 EXISTING SYSTEM**

Course and Certificate management (CACM) has many challenges such as nature of their institution providence and interest of the repudiate company head officer . Consequently traditional mechanisms that is certificate is made by paper and pen and also there is a cost for head office for pen and paper and alignment for printing it also be a cost .But the student or employee can get a certificate with alot of time, everything is containing a cost, but we can also getting a certificate.

**5.1.1 DISADVANTAGES IN EXISTING SYSTEM**

1.After complition of project student or employee to travel for certificate to get from institution(It may cause too travel cost)

2. waste of time ,money.

**5.2 PROPOSED SYSTEM**

Course and Certificate Management(CACM) is says about project in terms of proposed system is :here we are ready to make everything including registration and any kind information students or employees who are ready to join they can contact through our application without knowing the knowledge of our it they can see our information on web page, if they join in our IT we will provide them certificate through online, everything they can get from online. If employee or student interest they also sends a feedback to us , we ready to see this feedback of any time ensure it.

**5.2.1 Advantages of proposed system**

* user friendly
* low amount of maintains cost
* less time
* user friendly
* secure for your certificate
* no need of more number of employees
* easily available
* no need to communicate with staff for information.

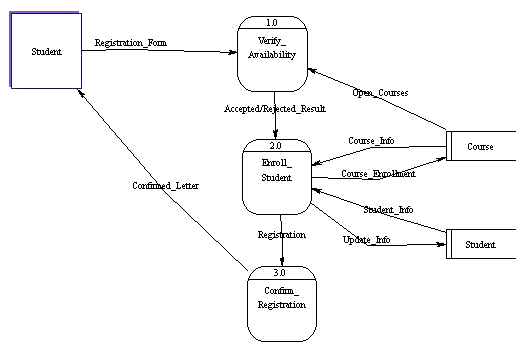
.

**5.3 UML DIAGRAMS**

**5.3.1 DATA FLOW DIAGRAM**

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

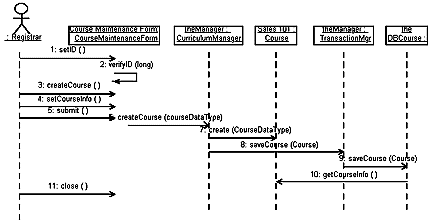
Course and Certificate management(CACM)is used for generate a certificate according to this data flow diagram, first employee or student he/she want to register after their registration, he can join the course. Now the registered persons data is stored in database which is seen only the admin , Here the database is sql and it is send a post message to the show data, before of this the admin should login the company website and he can generate a certificate to the students or employees by logging of that he can seen a show data by clicking on that, here table format can seen, here we can download a file it may be a pdf or XL, this can be seen in download list in your pc ,this can submit to the student or an employee.



***Figure 5.1 Data Flow Diagram***

**5.3.2 Sequence Diagram:**

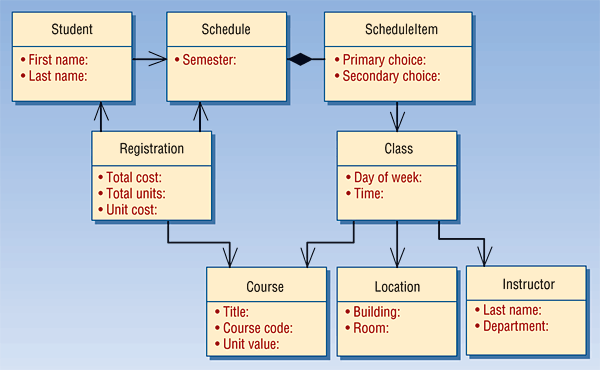
* A Sequence Diagram depicts the sequence of actions that occur in a this project i.e. Course and Certificate Management.
* It is a very useful tool to easily represent dynamic behavior of the system of this project i.e. CACM.
* The invocation of methods in each object and the order in which invocation occurs is captured in sequence diagram.
* So, sequence diagram represents the interactions among different objects of a system.



***Figure 5.2 Sequence diagram***

**5.3.3 Class Diagram:**

* A Class diagram shows a set of classes, interfaces and collaborations and their relationships.
* Class diagrams are important for constructing executable system through forward and reverse engineering apart from visualizing, specifying and documenting.
* Through class diagram, we model the static view which supports the functional requirements of a system.



***Figure 5.3 Class diagram***

**5.3.4 Use case Diagram:**

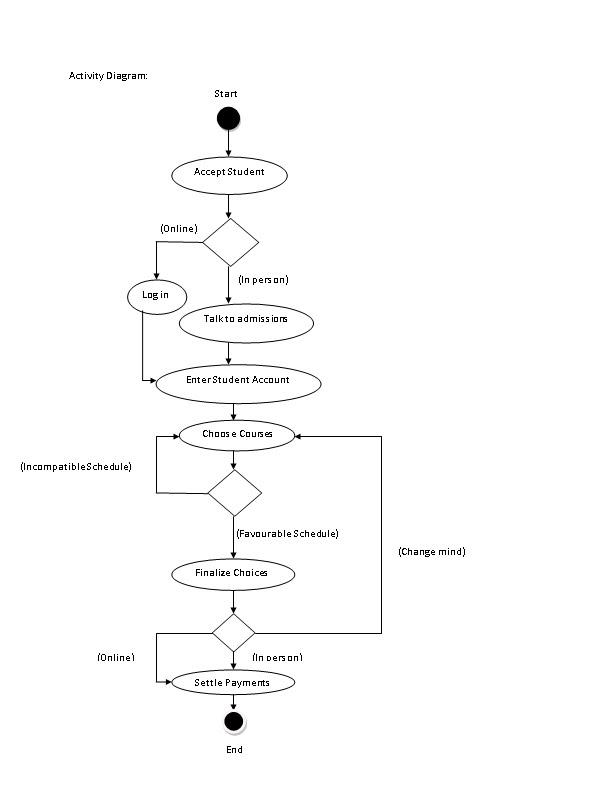
* A Use case diagram is a diagram that shows a set of usecases, actors and their relationships.
* Use case diagrams are central to modeling the behavior of a system, a subsystem, or a class.
* Each one shows a set of use cases and actors and their relationships.



***Figure 5.4 Use case diagram***

**5.3.5 Activity Diagram:**

* An Activity diagram illustrates the dynamic nature of a system by modeling the flow of control from activity to activity.
* Activity diagram consists a set of activities linked by transitions from one activity to other.
* Activity diagram can be used to model different aspects of a system.
* Activity diagrams show the overall flow of control.



***Figure 5.5 Activity diagram***

**CHAPTER 6**

**SYSTEM DESIGN**

**6.1 INPUT DESIGN**

The input design is the represented in the subline editor between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Models in the python in the case of .py extention for preparing input validations and steps to follow when error occur.
  + 1. **Objectives**

**1.** Input Design is the process of entering a data which is stored in a database form of my sql description of the input is entered by a student or employee. This design is important to noisy data in the project here input process and show the correct format for getting a certificate management and it shows correct information from the student or employee.

**2.** It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier , for easy registration done by a customer and to be free from noisy for any institution or any technology is easy understandable by seeing only by certificate. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

**3.** When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**6.2 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users. In output design it is determined how the information is to be displaced for immediate need and also it produces a hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

**1.** Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system only the adminer can use easily and effectively it provides to the student or employee. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

**2.** Select methods for presenting information.

**3.** Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities of joined course , current status or projections of the their project details.
* Future.
* showing the specific details of course joining information for a student or employee.
* showing data in a tabular format.

**CHAPTER 7**

**SYSTEM IMPLEMENTATION**

**7.1 IMPLEMENTATION PROCESS**

Implementation is the stage of the project when the theoretical design is turned out into a working procedure. Thus it can be considered to be the most critical stage in achieving a successful new system of any project and in giving the admin, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

**7.1.1 Subline text 3**

The subline text3 is an programming editor ,here we are using a python code as well as a in addition of that we writing a html , bootstrap , cascade style sheet ,java query

.Coming to python , here we are seen a various number of modules where each model having a interconnection with .html files which are encapsulates with their

own data and everything in various sub ordinates. The html code which are placed in their templates, and images also. it can be run by setting a path on setting.py here we want to do overall operating system pattern. And we can also storing a database which is used by my sql here what are the data we prepared a code for html in subline which is made a connection by running a server which is Xampp and execute it before of that in database we want to create it a consistent data ,here we can also add a data in database very thing is included. Thus about subline.

**7.1.2 DATA BASE(MY SQL)**

In this database ,here we used for storing a data, the data which is in the form of

field values because here we are entering a data which are should be unique so we need this kind of data , there be a large amount of storing a information which are be enroll for selected on time to time .There are large number of people are interest to joined a course , in this issue we need a secure data to store in thier enrollment included.

**7.1.3 Source Node**

We written a many lines of source code which are in various formats for front end source code is different format and back end source code is different which are in various format, but the logic code is very important, what we represent in the logic code is base root for the entier program ,

**7.1.4 Django from for python**

**Django** is a free and open source web application framework written in Python. A framework is nothing more than a collection of modules that make development easier. They are grouped together, and allow you to create applications or websites from an existing source, instead of from scratch. Here in this project it includes various number of modules are present these are embeded in its for running a python project.

**7.1.5 Server connection**

For running a project we want to run the server , before of that we want to install the XAMPP server after installation we want to open the XAMPP control panel ,here we want to start the apache and sql , these are ready to start to execute the process without the help of XAMPP control panel we can't run the project but we can run by using any of the server.

**CHAPTER 8**

**SYSTEM TESTING**

Testing is a process of finding faults with the package. To test the software there are several testing methodologies.

**Testing Objectives**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

* + Testing is a process of executing a program with the intent of finding an error.
  + A successful test is one that uncovers an as yet undiscovered error.
  + A good test case is one that has a high probability of finding error, if it exists.
  + The tests are inadequate to detect possibly present errors.
  + The software more or less confirms to the quality and reliable standards.

**Levels of Testing**

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below…



Client Needs

Requirements

Design

Code

**Types of Testing**

* Unit Testing
* System Testing
* User Input Validation Testing

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Selection, Listing, Execution, and Import & Export. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error.

**System Testing**

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal to see if software meets its requirements

**User input validation**

The User input must be validated to confirm to expected values. The fields should also not be empty.

**7.2. TEST CASES**

**Test case 1:**

|  |  |  |
| --- | --- | --- |
| **1.** | **Test case ID** | **Student / employee entering details** |
| **2.** | **Precondition** | * Enter name * Enter college /company name * Enter given courses, DOB * Enter submit button |
| **3.** | **Description** | Donor has to enter details. |
| **4.** | **Test Steps** | * Enter name * Enter college /company name * Enter given courses, DOB * Enter submit button |
| **5.** | **Expected Output** | If the details are entered into the database, it has to give the status as successfully registered. |
| **6.** | **Actual Output** | Toast with “Successfully Registered” displayed. |
| **7.** | **Status** | **YES** |
| **8.** | **Remarks** | None |

**Table 7.1 Test case 1**

**Test case 2**

|  |  |  |
| --- | --- | --- |
| **1.** | **Test case ID** | **Login** |
| **2.** | **Precondition** | * Enter name * Enter password |
| **3.** | **Description** | If login and password match, the users (donor or recipient) respective pages have to be displayed |
| **4.** | **Test Steps** | * Enter name * Enter Password * Click “login” button |
| **5.** | **Expected Output** | Update page for admin and search page for recipient will be opened. |
| **6.** | **Actual Output** | It shows the data who are entered of registered it displays table , generates a certificate as displayed. |
| **7.** | **Status** | **YES** |
| **8.** | **Remarks** | None |

**Table. 7.2 Test Case 2**

**Test case 3**

|  |  |  |
| --- | --- | --- |
| **1.** | **Test case ID** | **Login failure** |
| **2.** | **Precondition** | * Enter wrong ID. * Enter wrong Password. |
| **3.** | **Description** | If the ID and password mismatch, a error message is displayed |
| **4.** | **Test Steps** | * Enter login id * Enter Password * Click “login” button |
| **5.** | **Expected Output** | Invalid admin ,message has to be displayed. |
| **6.** | **Actual Output** | Displays a message saying ”Invalid User name/Password” |
| **7.** | **Status** | **YES** |
| **8.** | **Remarks** | None |

**Table. 7.3 Test Case 3**

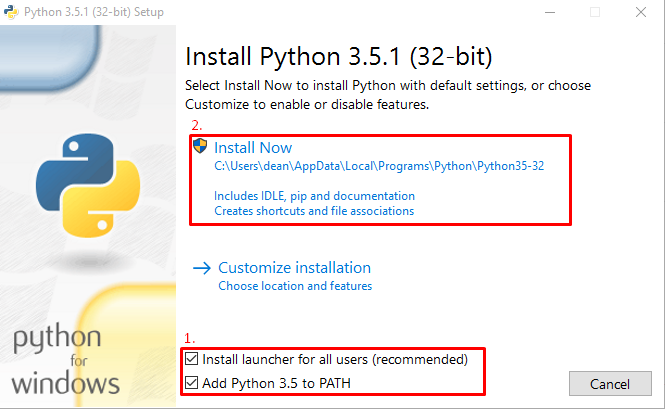
**Test Case 4**

|  |  |  |
| --- | --- | --- |
| **1.** | **Test case ID** | **Searching and generating results** |
| **2.** | **Precondition** | * Enter student / employee correct details. * Choose the correct course for the employee / student. |
| **3.** | **Description** | After entering login details and , the registered students or employees about their registered field values are shown in shown data option .Here the admin can download a pdf as well as XL file. |
| **4.** | **Test Steps** | * Enter the login details * Choose the valid mail id . |
| **5.** | **Expected Output** | Course Certificate has to be generated. |
| **6.** | **Actual Output** | After the submission data has to be generated. |
| **7.** | **Status** | **YES** |
| **8.** | **Remarks** | None |

**Table. 7.4 Test Case 4**

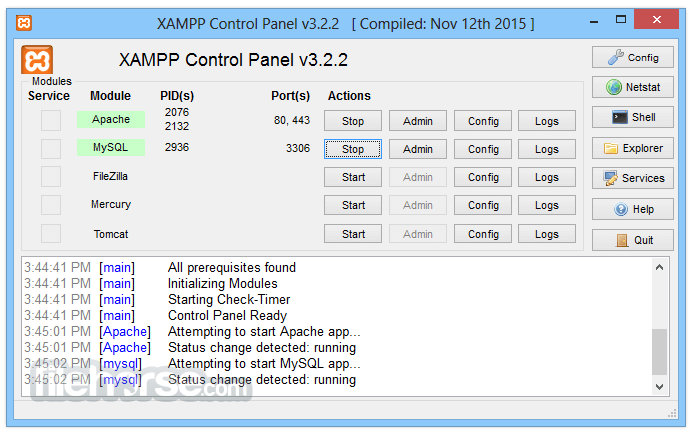
**CHAPTER 9**

**SCREEN LAYOUTS**



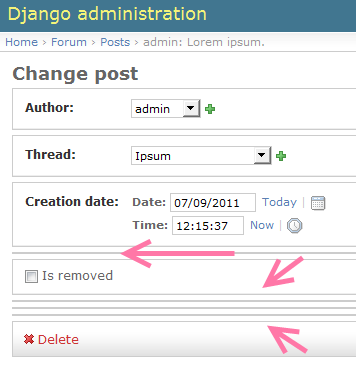
***Screen 9.1 python installation***

In above screen shot shows that the structure of python install set up and there is a bit and more information is present in it , it estimates the python environment setup to ensure the process and to allocate their own needs to setup. Here we are enrolls the overall information of the python environment setup and to be set a path to all the operating system path and to associate to their own needs to set up to enclourse the specific needs to develope their needs.

***.***

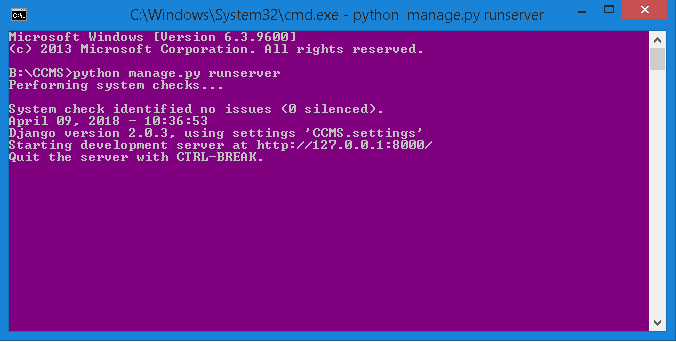
***Screen 9.2 Lets start XAMPP Control Panel***

The above screen shot shows that making a connection to the project by only the help of XAMPP control panel by clicking the XAMPP server and starting the actions .



***Screen 9.3 Django frame work for python***

The above screen shot shows that to develope a web application by using django frame work it shows the how many people want to add in my company.

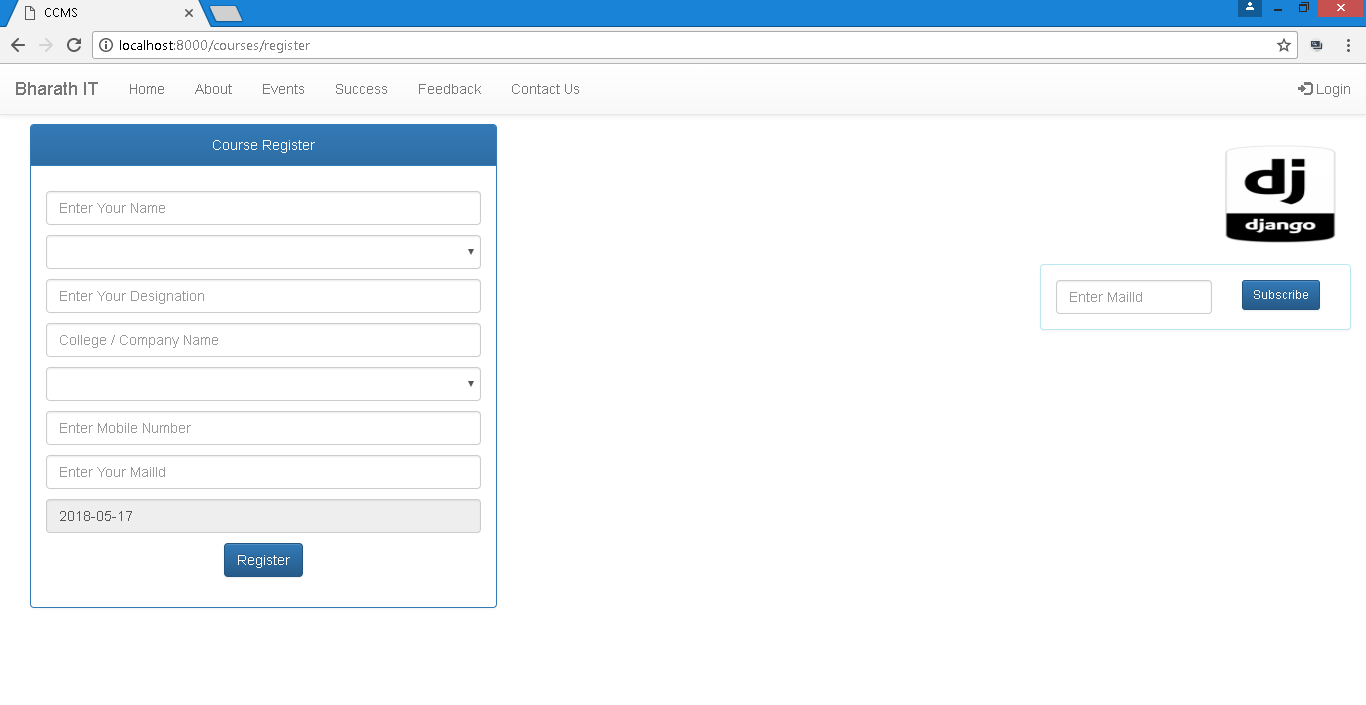


***Screen 9.4CMD for running python***

The above screen shot shows that how to run the python code on command prompt and to be set ,before of that we want to do a migrations for entier programs.

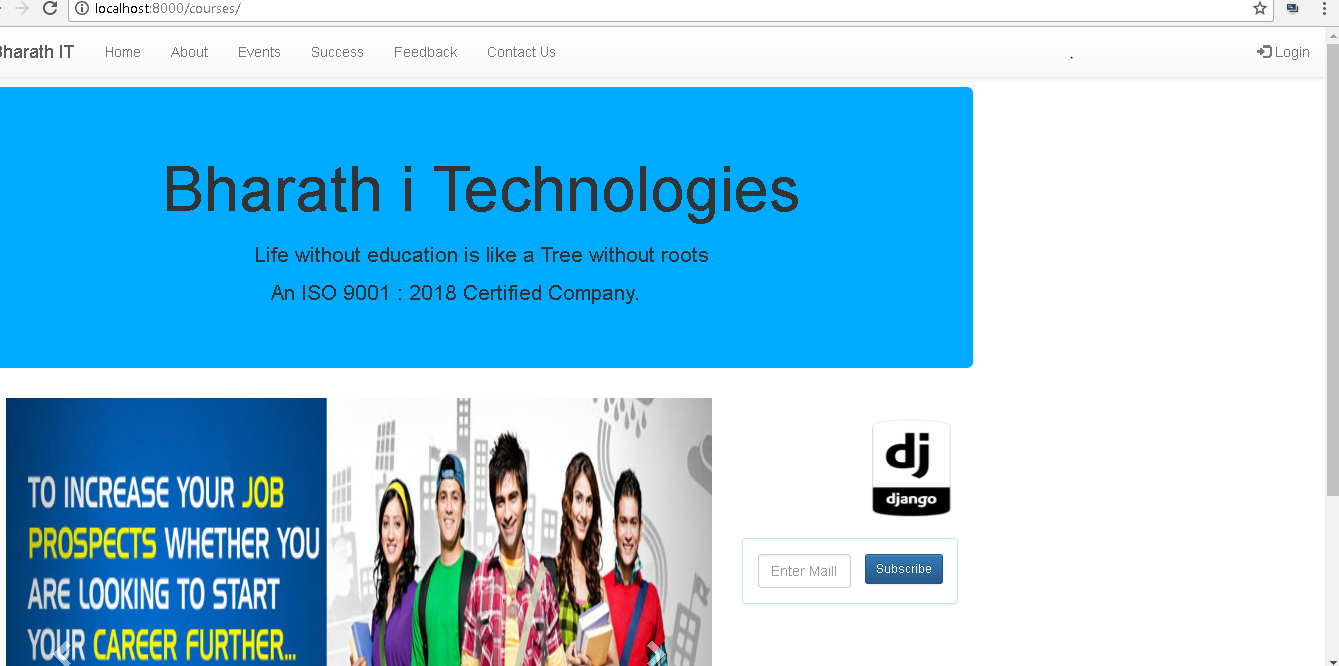
***Screen 9.5 Registration process***

The below screen shot shows that to make a registration first , it is need for every employee and student it is compulsory to join a course in my institiute.



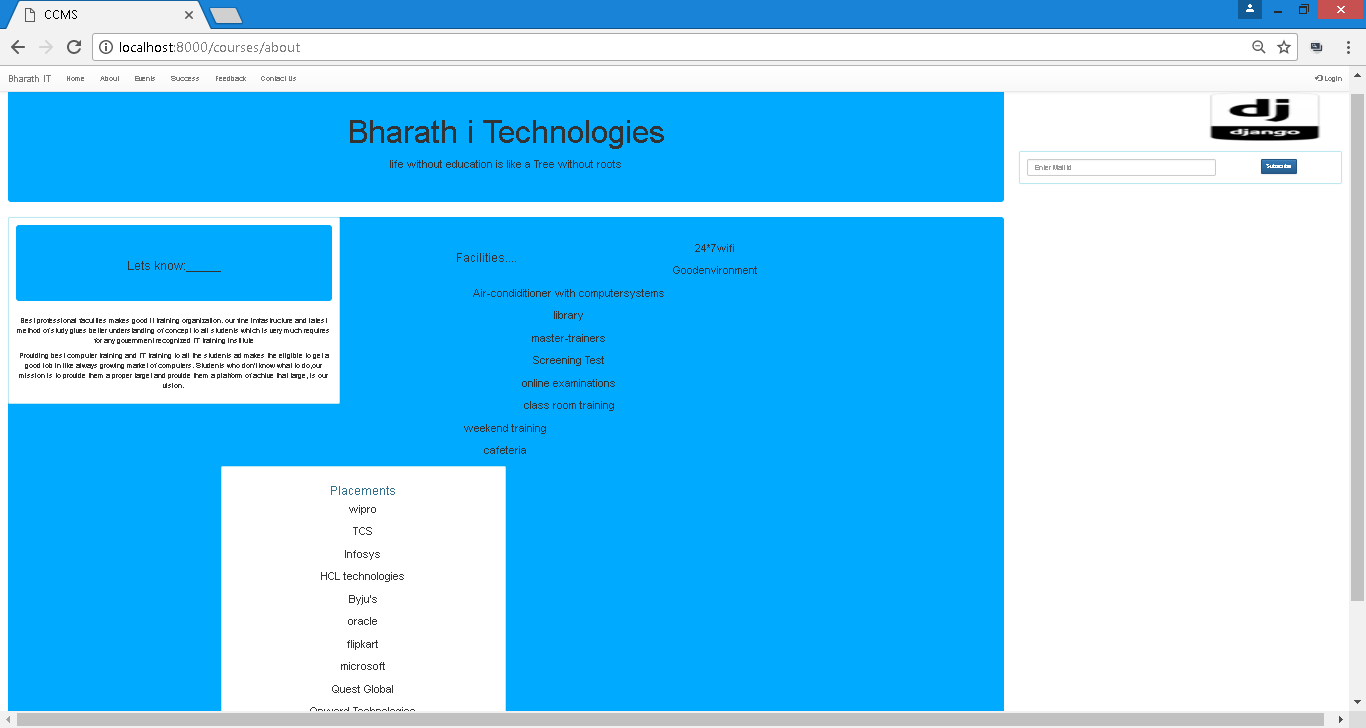
***Screen 9.6 CACM Home page***

The below screen shows that , This shows a home page for our institute it contains our company images and subscribe option to know overall data.

******

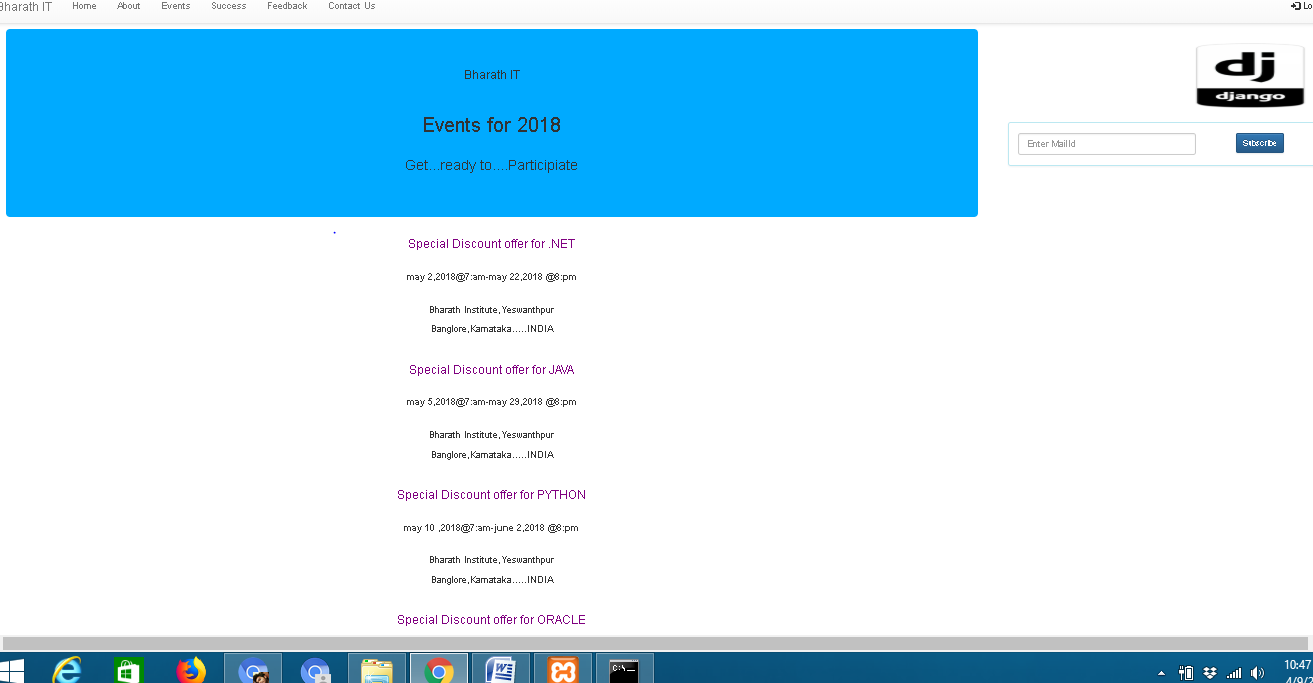
***Screen 9.7 CASM ABOUT***

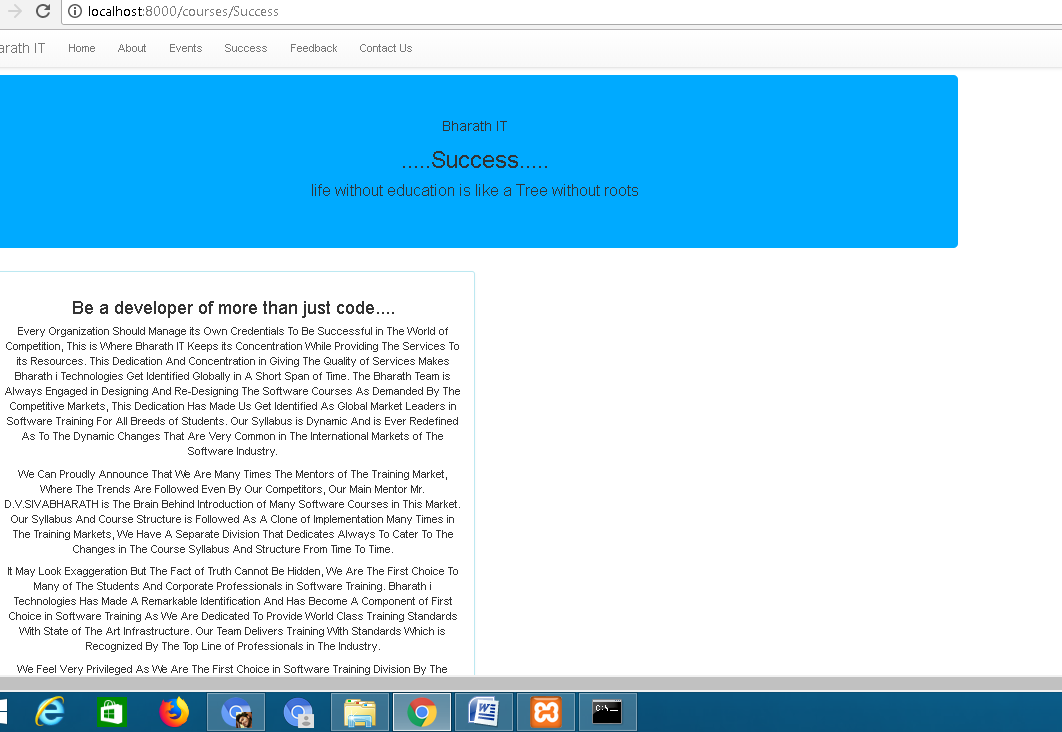
The below screen shot shows that know about company /institute details which are in clear in manner. We include all the facilities and placements data.



***Screen 9.8 CACM Events***

The below screen shot shows that to know the overall events what are making in future for understanding the costumers which are student or employee for knowing more languages it depends upon their interest .



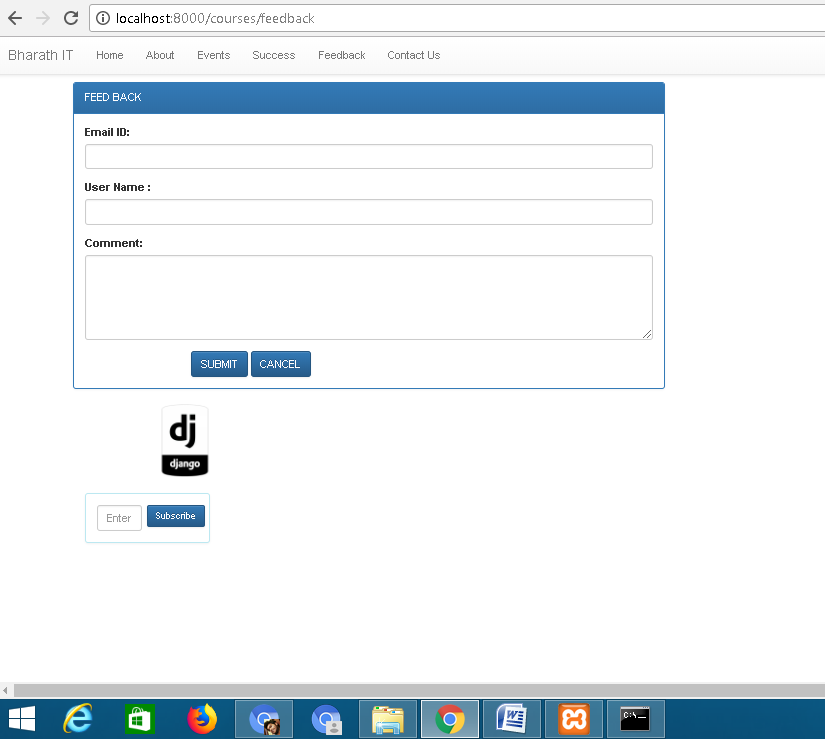
****

***Screen 9.9 CACM Success***

The above screen shot shows that to what are the success stories are happen in our company by getting a various awards by the reputted states in our country.

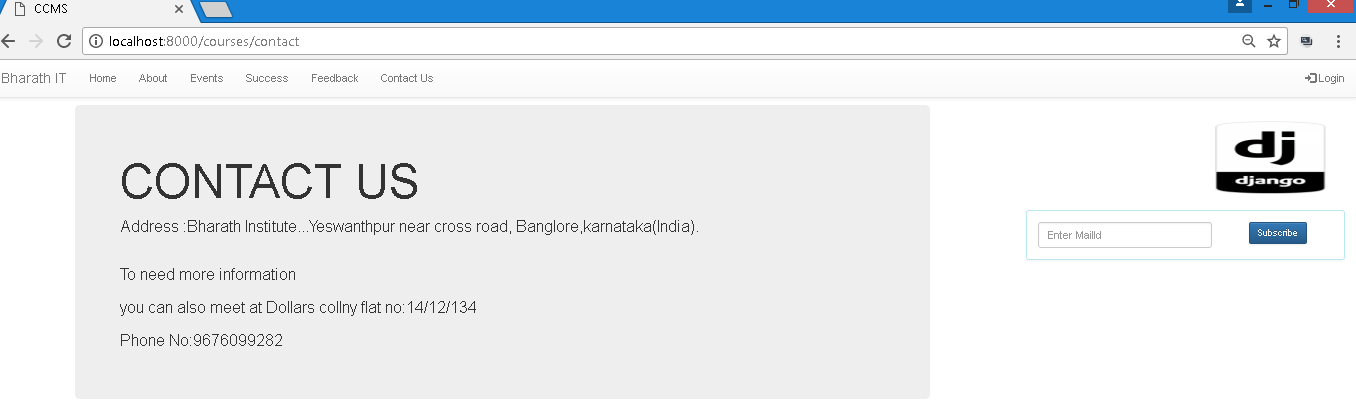
***Screen 9.10 CACM Feedback***

*The customers which are participate in the company for joining purpose , if they want to say any information through online instead of that they will face shye while to trainer, we introduced Feedback , we get a information from employees or students , we see the data in the database , its only possible by the admin. The admin can read all the data then find failures, later discussion of that we notify the problems and ready to make solutions for the students or employees.*

******

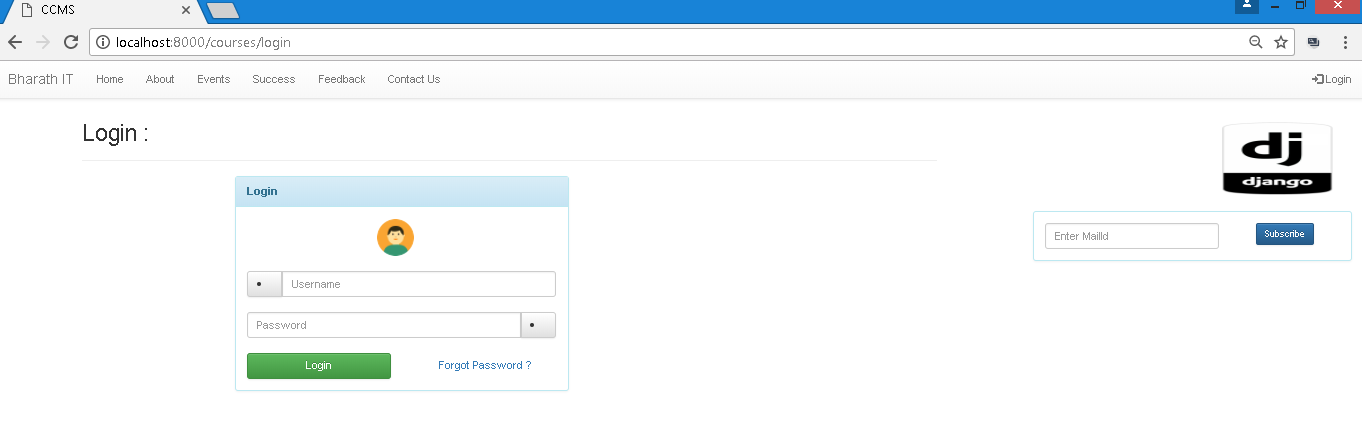
***Screen 9.11 CACM for Contact us***

The below screen shot shows that ,overall information of our company address is visible for the employees or students or, the people who searched about contact address of our comapny.



***Screen 9.12 CACM for login***

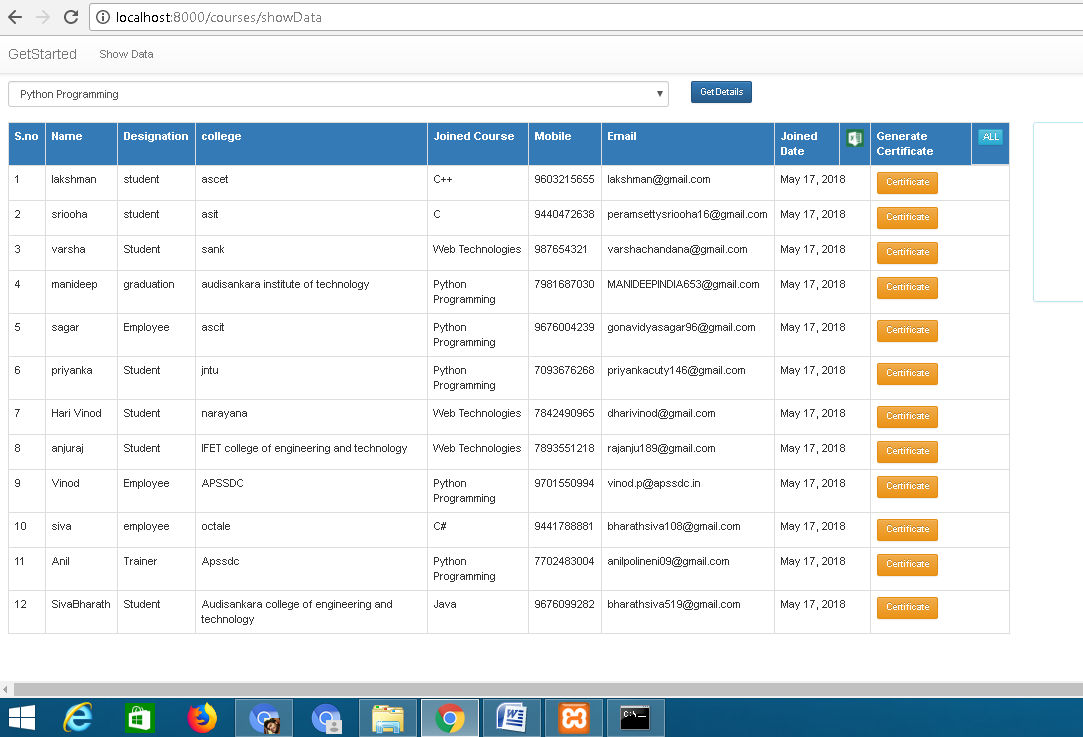
By knowing the authorized person of an employee in our company only that should be make a secure login id and password for making strong security.



***Screen 9.13 CACM for Shown Data***

The below screen shot shows that ,required output data for an employee or student what are the required fields entered by the student or employee is same as registered data is seen in tabular format , the data is encapsulate to seen in the certificate. Here we have two options one is pdf and another one is XML file. We can also get both. If we want to need only specific course registration is also possible.

Those are registered students and employees overall number of people data can get a certificate by clicking a all.



.

**CHAPTER 10**

**SAMPLE SOURCE CODE**

**about.html**

{% extends 'base.html' %}

{% load staticfiles %}

{% block content %}

<style>

.type-black{color:black;}

.bg-blue{background-color:#00aaff;}

</style>

<div class="jumbotron text-center jumbotron bg-blue">

<h1>Bharath i Technologies<h1>

<p>life without education is like a Tree without roots</p>

</div>

<div class="col-sm-4 text-center panel panel-body panel-info">

<style>

.type-black{color:black;}

.bg-blue{background-color:#00aaff;}

</style>

<div class="jumbotron text-center jumbotron bg-blue">

<h3>Lets know:\_\_\_\_\_</h3>

</div>

<p>Best professional faculties makes good it training organization. our fine infrastructure and latest method of study gives better understanding of concept to all students which is very much requires for any government recognized IT training institute</p>

<p>Providing best computer training and IT training to all the students ad makes the eligible to get a good job in like always growing market of computers. Students who don't know what to do,our mission is to provide them a proper target and provide them a platform of achive that targe, is our vision.</p>

</div>

<style>

.type-black{color:black;}

.bg-blue{background-color:#00aaff;}

</style>

<div class="jumbotron text-center jumbotron bg-blue">

<div class="container text-black">

<div class="row">

<div class="col-sm-6">

<h3>Facilities....</h3>

</div>

<p>24\*7wifi</p>

<p>Goodenvironment</p>

<p>Air-condiditioner with computersystems</p>

<p>library</p>

<p> master-trainers</p>

<p>Screening Test</p>

<p>online examinations</p>

<p>class room training</p>

<p>weekend training</p>

<p>cafeteria</p>

</div>

<div class="col-sm-6 text-center panel panel-body panel-info">

<h3 class="text-info">Placements</h3>

<p>wipro</p>

<p>TCS</p>

<p>Infosys</p>

<p>HCL technologies</p>

<p>Byju's</p>

<p>oracle</p>

<p>flipkart</p>

<p>microsoft</p>

<p>Quest Global</p>

<p>Onward Technologies</p>

<p>3i infotech</p>

<p>Hexaware technologies</p>

</div>

</div>

</div>

{% endblock %}

**Contact.html**

{% extends 'base.html' %}

{% block content %}

<div class="container">

<div class="jumbotron">

<h1>CONTACT US</h1>

<p>Address :Bharath Institute...Yeswanthpur near cross road, Banglore,karnataka(India).</p>

<br>

<p>To need more information</p>

<p>you can also meet at Dollars collny flat no:14/12/134</p>

<p>Phone No:9676099282</p>

</div>

</div>

{% endblock %}

**error.html**

{% extends 'base.html' %}

{% load staticfiles %}

{% block content %}

<div class="container">

<div class="center-block">

<a class="btn btn-default btn-xs glyphicon glyphicon-circle-arrow-left" href="{% url 'joinCourse:courseReg' %}"></a>

<h1 class="text-danger">ERROR Try Again Later...!</h1><br>

<img src="{% static 'Images/confuse1.png' %}" style="height: 250px;width: 50%" class="img-responsive">

</div>

</div>

{% endblock %}

events.html

{% extends 'base.html' %}

{% load staticfiles %}

{% block content %}

<style>

.type-black{color:black;}

.bg-blue{background-color:#00aaff;}

.purple-text{color: purple}

</style>

<div class="container-fluid ">

<div class="jumbotron text-center jumbotron bg-blue">

<h4>Bharath IT</h4>

<br>

<h2>Events for 2018</h2><br>

<p>Get...ready to....Participiate </p>

</div>

<div class="text-center">

<h4 class="purple-text"><p>Special Discount offer for .NET</p></h4><br>

<b><p>may 2,2018@7:am-may 22,2018 @8:pm</p></b><br>

<p>Bharath Institute,Yeswanthpur</p>

<p>Banglore,Karnataka.....INDIA</p>

<br>

<h4 class="purple-text"><p>Special Discount offer for JAVA</p></h4><br>

<b><p>may 5,2018@7:am-may 29,2018 @8:pm</p></b><br>

<p>Bharath Institute,Yeswanthpur</p>

<p>Banglore,Karnataka.....INDIA</p>

<br>

<h4 class="purple-text"><p>Special Discount offer for PYTHON</p></h4><br>

<b><p>may 10 ,2018@7:am-june 2,2018 @8:pm</p></b><br>

<p>Bharath Institute,Yeswanthpur</p>

<p>Banglore,Karnataka.....INDIA</p>

<br>

<h4 class="purple-text"><p>Special Discount offer for ORACLE</p></h4><br>

<b><p>may 20,2018@7:am-june 15,2018 @8:pm</p></b><br>

<p>Bharath Institute,Yeswanthpur</p>

<p>Banglore,Karnataka.....INDIA</p>

<br>

<h4 class="purple-text"><p>Special Discount offer for SALESFORCE</p></h4><br>

<b><p>june 2,2018@7:am-june 22,2018 @8:pm</p></b><br>

<p>Bharath Institute,Yeswanthpur</p>

<p>Banglore,Karnataka.....INDIA</p>

</div>

</div>

{% endblock %}

Feedback.html

{% extends 'base.html' %}

{% load staticfiles %}

{% block content %}

<!DOCTYPE html>

<html>

<head>

<title>FEED BACK</title>

</head>

<body>

<form>

<div class="container col-sm-8 ">

<div class="panel panel-primary col-sm-offset-1">

<div class="panel-heading">FEED BACK</div>

<div class="panel-body">

<div class="form-group">

<label for="email">Email ID:</label>

<input type="email" class="form-control" id="email">

</div>

<div class="form-group">

<label for="UserName">User Name :</label>

<input type="UserName" class="form-control" id="UserName">

</div>

<div class="form-group">

<label for="comment">Comment:</label>

<textarea class="form-control" rows="5" id="comment"></textarea>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-primary">SUBMIT</button>

<button type="reset" class="btn btn-primary">CANCEL</button>

</div>

</div>

</form>

</body>

</html>

{% endblock %}

Register.html

{% extends 'base.html' %}

{% load staticfiles %}

{% block content %}

<style>

.mt{margin-top: 10px;}

</style>

<div class="container-fluid col-sm-6">

<form action="{% url 'joinCourse:RegisterHere' %}" method="post">

{% csrf\_token %}

<div class="panel panel-primary center-block">

<div class="panel-heading text-center">Course Register</div>

<div class="panel-body">

<div class="row form-group">

<div class="col-sm-12 mt">

<input type="text" name="name" id="name" class="form-control" placeholder="Enter Your Name" required="required">

</div>

<div class="col-sm-12 mt">

<select name="gender" class="form-control" placeholder="gender" required="required">

<option></option>

<option>Male</option>

<option>Female</option>

</select>

</div>

<div class="col-sm-12 mt">

<input type="text" name="design" class="form-control" placeholder="Enter Your Designation" required="required">

</div>

<div class="col-sm-12 mt">

<input type="text" name="college" class="form-control" placeholder="College / Company Name" required="required">

</div>

<div class="col-sm-12 mt">

<select name="courses" class="form-control" required="required">

<option></option>

<option>C</option>

<option>C++</option>

<option>Python Programming</option>

<option>C#</option>

<option>Java</option>

<option>PHP</option>

<option>Web Technologies</option>

</select>

</div>

<div class="col-sm-12 mt">

<input type="text" name="mobile" class="form-control" placeholder="Enter Mobile Number" required="required">

</div>

<div class="col-sm-12 mt">

<input type="text" name="email" class="form-control" placeholder="Enter Your MailId" required="required">

</div>

<div class="col-sm-12 mt">

<input type="text" name="date" id="datepickers" class="form-control" placeholder="Join Date" value="2018-05-17" readonly="readonly" required="required">

</div>

<div class="col-sm-12 mt">

<input type="submit" class="center-block btn btn-primary" value="Register">

</div>

</div>

</div>

</div>

</form>

</div>

<script src="{% static 'scripts/datePicker/jquery-1.12.4.js' %}"></script>

<script src="{% static 'scripts/datePicker/jquery-ui.js' %}"></script>

<script>

$( function() {

$( "#datepicker" ).datepicker({

changeMonth: true,

changeYear: true

});

} );

</script>

views.py

. from django.shortcuts import render,redirect

from joinCourse.models import Register,employee

from io import BytesIO

from django.views.generic import View

from django.template.loader import get\_template

from django.template import loader

#from reportlab.pdfgen import canvas

from django.http import HttpResponse,Http404,HttpResponseRedirect

from reportlab.pdfgen import canvas

from reportlab.lib.units import inch,mm

from reportlab.lib.pagesizes import letter,landscape

from fpdf import FPDF

from django.contrib.auth.models import User

from django.contrib.auth import authenticate,login as dj\_login,logout

from django.contrib.auth.decorators import login\_required

from django.core.mail import send\_mail

# from .utils import render\_to\_pdf

#return HttpResponseRedirect('/url-where-you-want-to-redirect/')

# from django.core.urlresolvers import reverse

from django.contrib.messages import constants as messages

MESSAGE\_TAGS = {

messages.INFO: '',

50: 'critical',

}

import logging

logging.basicConfig(level=logging.INFO)

logger = logging.getLogger(\_\_name\_\_)

def subscribe(request):

mail=request.POST['subscribe']

# fromMail="bharathsiva519@gmail.com"

fromMail="alex.apssdc@gmail.com"

subject="Get Started Courses"

body="Thank and Regards,We Will Get Back to You."

send\_mail(subject,body,fromMail,[mail],fail\_silently=False)

return render(request,"joinCourse/thankYou.html")

def home(request):

return render(request,"joinCourse/home.html")

def about(request):

return render(request,"joinCourse/about.html")

def events(request):

return render(request,"joinCourse/events.html")

def Success(request):

return render(request,"joinCourse/Success.html")

def feedback(request):

return render(request,"joinCourse/feedback.html")

def contact(request):

return render(request,"joinCourse/contact.html")

def courseReg(request):

return render(request,"joinCourse/register.html")

def RegisterHere(request):

try:

register=Register(name=request.POST['name'],gender=request.POST['gender'],designation=request.POST['design'],college=request.POST['college'],course=request.POST['courses'],mobile=request.POST['mobile'],email=request.POST['email'],joined\_date=request.POST['date'])

register.save()

return render(request,"joinCourse/thankYou.html")

except Exception as e:

print(e)

return render(request,"joinCourse/error.html")

def getDetails(request):

course=request.POST['course']

tech=Register.objects.filter(course=course)

return render(request,"joinCourse/joinedCourse.html",{'tech':tech})

class GenerateOne(View):

def get(self,request,stu\_id):

register=Register.objects.filter(id=stu\_id)

response = HttpResponse(content\_type='application/pdf')

response['Content-Disposition'] = 'attachment; filename="IndividualCourse.pdf"'

html="<h1>Success</h1>"

c = canvas.Canvas(response,pagesize=letter)

width, height = letter

for r in register:

c.drawImage("icons/certificate.jpg", 0,0, 8.5\*inch,10\*inch)

c.setFont('Helvetica', 12)

course=r.course

c.drawString(270, 450,course.upper())

name=r.name

c.drawString(270, 300, name.upper())

date=r.joined\_date

c.drawString(150, 200, str(date))

c.showPage()

c.save()

return response

class GenerateAll(View):

def get(self,request):

register=Register.objects.all()

#template =get\_template("joinCourse/home.html")

response = HttpResponse(content\_type='application/pdf')

response['Content-Disposition'] = 'attachment; filename="Course.pdf"'

html="<h1>Success</h1>"

c = canvas.Canvas(response,pagesize=letter)

width, height = letter

pdf=FPDF()

for r in register:

c.drawImage("icons/certificate.jpg", 0,0, 8.5\*inch,10\*inch)

c.setFont('Helvetica', 12)

course=r.course

c.drawString(270, 450,course.upper())

name=r.name

c.drawString(270, 300, name.upper())

date=r.joined\_date

c.drawString(150, 200, str(date))

c.showPage()

c.save()

return response

def login(request):

return render(request,"joinCourse/login.html")

def logins(request):

if request.user.is\_authenticated:

return render(request,"dashBoard.html")

if request.method=='POST':

username=request.POST.get('username')

password=request.POST.get('password')

user=authenticate(username=username,password=password)

if user is None:

return render(request,"joinCourse/LoginError.html")

else:

dj\_login(request,user)

refurl=request.GET.get('next')

if refurl is not None:

return redirect(refurl)

return render(request,"dashBoard.html")

else:

return render(request,"joinCourse/login.html")

def logouts(request):

if request.user.is\_authenticated:

logout(request)

return render(request,"joinCourse/login.html")

else:

return render(request,"joinCourse/login.html")

return render(request,"joinCourse/login.html")

@login\_required()

def showData(request):

register=Register.objects.all().order\_by('id').reverse()

context={'register':register}

return render(request,"joinCourse/joinedCourse.html",context)

# def uploadData(request):

# data = {}

# if request.method=="GET":

# return render(request, "joinCourse/uploadForm.html", data)

# # if not GET, then proceed

# try:

# csv\_file = request.FILES["csv\_file"]

# print(csv\_file)

# except Exception as e:

# print(e)

# # if not csv\_file.name.endswith('.csv'):

# # messages.error(request,'File is not CSV type')

# # return HttpResponseRedirect(reverse("myapp:upload\_csv"))

# # #if file is too large, return

# # if csv\_file.multiple\_chunks():

# # messages.error(request,"Uploaded file is too big (%.2f MB)." % (csv\_file.size/(1000\*1000),))

# # return HttpResponseRedirect(reverse("myapp:upload\_csv"))

# # file\_data = csv\_file.read().decode("utf-8")

# # lines = file\_data.split("\n")

# # #loop over the lines and save them in db. If error , store as string and then display

# # for line in lines:

# # fields = line.split(",")

# # data\_dict = {}

# # data\_dict["name"] = fields[0]

# # data\_dict["start\_date\_time"] = fields[1]

# # data\_dict["end\_date\_time"] = fields[2]

# # data\_dict["notes"] = fields[3]

# # try:

# # form = EventsForm(data\_dict)

# # if form.is\_valid():

# # form.save()

# # else:

# # logging.getLogger("error\_logger").error(form.errors.as\_json())

# # except Exception as e:

# # logging.getLogger("error\_logger").error(repr(e))

# # pass

# # except Exception as e:

# # logging.getLogger("error\_logger").error("Unable to upload file. "+repr(e))

# # messages.error(request,"Unable to upload file. "+repr(e))

# # return HttpResponseRedirect(reverse("myapp:upload\_csv"))

# return render(request,"joinCourse/uploadForm.html")

def uploadData(request):

data = {}

if "GET" == request.method:

return render(request, "joinCourse/uploadForm.html", data)

# if not GET, then proceed

try:

csv\_file = request.FILES["csv\_file"]

if not csv\_file.name.endswith('.csv'):

messages.error(request,'File is not CSV type')

return HttpResponseRedirect(reverse("courses:uploadData"))

#if file is too large, return

if csv\_file.multiple\_chunks():

messages.error(request,"Uploaded file is too big (%.2f MB)." % (csv\_file.size/(1000\*1000),))

return HttpResponseRedirect(reverse("courses:uploadData"))

file\_data = csv\_file.read().decode("utf-8")

lines = file\_data.split("\n")

#loop over the lines and save them in db. If error , store as string and then display

for line in lines:

fields = line.split(",")

data\_dict = {}

data\_dict["name"] = fields[0]

data\_dict["start\_date"] = fields[1]

data\_dict["end\_date"] = fields[2]

data\_dict["notes"] = fields[3]

try:

form = employee(data\_dict)

if form.is\_valid():

form.save()

else:

logging.getLogger("error\_logger").error(form.errors.as\_json())

except Exception as e:

logging.getLogger("error\_logger").error(repr(e))

pass

except Exception as e:

logging.getLogger("error\_logger").error("Unable to upload file. "+repr(e))

#messages.error(request,"Unable to upload file. "+repr(e))

return HttpResponse("Success")

# def login1(request):

# next=request.GET.get('next','/')

# if request.method=="POST":

# username=request.POST['username']

# password=request.POST['password']

# user=authenticate(username=username,password=password)

# if user is not None:

# if user.is\_active:

# login(request,user)

# return HttpResponseRedirect('/')

# else:

# HttpResponse("Inactive user!")

# else:

# return HttpResponseRedirect('/login/')

# return render(request,"joinCourse/login.html",{'redirect\_to':next})

# p.showPage()

# p.save()

# pdf = buffer.getvalue()

# buffer.close()

# response.write(pdf)

#canvas.Canvas(buffer)

#return response #HttpResponse(pdf.output("PDF/a.pdf","F"),open("PDF/a.pdf"))

# def ShowOne(request,stu\_id):

# register=Register.objects.filter(id=stu\_id)

# context={'registerOne':register}

# return render(request,"joinCourse/joinedCourse.html",context)

# def ge(request):

# return pdf.gens()

# def genPDF(request):

# # Create the HttpResponse object with the appropriate PDF headers.

# response = HttpResponse(content\_type='application/pdf')

# response['Content-Disposition'] = 'attachment; filename="somefilename.pdf"'

# # Create the PDF object, using the response object as its "file."

# p = canvas.Canvas(response)

# # Draw things on the PDF. Here's where the PDF generation happens.

# # See the ReportLab documentation for the full list of functionality.

# p.drawString(1, 1, "Hello world.")

# # Close the PDF object cleanly, and we're done.

# p.showPage()

# p.save()

# return response

#from django.core.files.storage import FileSystemStorage

#from fpdf import FPDF

#from django.template.loader import render\_to\_string

#from weasyprint import HTML, CSS

# Create your views here.

# def genPDF(request):

# register=Register.objects.all()

# pname='newBatch'

# if register!=None:

# pdf=FPDF()

# for register in register:

# print(register.name)

# pdf.add\_page("a4")

# pdf.set\_font('Arial', 'B', 16)

# pdf.cell(90, 10, register.name,1,0,"C")

# pdf.cell(90, 10, register.course,1,0,"C")

# #return render(request,"retrive.html",context)

# return pdf.output("static/PDFs/"+pname+".pdf","F")

# else:

# return HttpResponse("Something Went Wrong...!")

# return render(request,"retrive.html",{'register':register})

# 0

# def genPDF(request):

# try:

# paragraphs = ['first paragraph', 'second paragraph', 'third paragraph']

# html\_string = render\_to\_string('core/pdf\_template.html', {'paragraphs': paragraphs})

# html = HTML(string=html\_string)

# fs = FileSystemStorage('/PDFs')

# with fs.open('mypdf.pdf') as pdf:

# response = HttpResponse(pdf, content\_type='application/pdf')

# response['Content-Disposition'] = 'attachment; filename="mypdf.pdf"'

# return response

# return html.write\_pdf(target='/PDFs/mypdf.pdf')

# except Exception as e:

# print(e)

# class GenerateOne(View):

# def get(self,request,stu\_id):

# register=Register.objects.filter(id=stu\_id)

# pdf=FPDF()

# for register in register:

# att=register.name+'\_'+str(register.joined\_date)+'\_'+register.course

# pdf.add\_page("a4")

# pdf.image('icons/certificate.jpg', 10, 8, 280)

# pdf.set\_font('Arial', 'B', 16)

# pdf.ln(70)

# pdf.cell(90, 10, '',0,0,"C")

# course=register.course

# pdf.cell(100, 10,course.upper(),0,0,"C")

# pdf.ln(40)

# pdf.cell(90, 10, '',0,0,"C")

# name=register.name

# pdf.cell(100,10,name.upper(),0,0,"C")

# pdf.ln(20)

# pdf.cell(60, 10, '',0,0,"C")

# date=register.joined\_date

# pdf.cell(40, 10,str(date),0,0,"C")

# my\_data=pdf.output("PDF/"+att+".pdf","F")

# response = HttpResponse(my\_data, content\_type='application/pdf')

# response['Content-Disposition'] = 'attachment; filename="foo.pdf"'

# return HttpResponse(response)

# class GenerateAll(View):

# def get(self,request):

# register=Register.objects.all()

# #template =get\_template("joinCourse/home.html")

# response = HttpResponse(content\_type='application/pdf')

# response['Content-Disposition'] = 'attachment; filename="Course.pdf"'

# html="<h1>Success</h1>"

# c = canvas.Canvas(response)

# pdf=FPDF()

# for r in register:

# # print(r.name)

# # pdf.add\_page("a4")

# # pdf.image('icons/certificate.jpg', 10, 8, 280)

# # pdf.set\_font('Arial', 'B', 16)

# # pdf.ln(70)

# # pdf.cell(90, 10, '',0,0,"C")

# # course=r.course

# # pdf.cell(100, 10,course.upper(),0,0,"C")

# # pdf.ln(40)

# # pdf.cell(90, 10, '',0,0,"C")

# # name=r.name

# # pdf.cell(100,10,name.upper(),0,0,"C")

# # pdf.ln(20)

# # pdf.cell(60, 10, '',0,0,"C")

# # date=r.joined\_date

# # pdf.cell(40, 10,str(date),0,0,"C")

# c.setFont('Helvetica', 12)

# c.drawString(40, 10, str(date))

# c.showPage()

# c.save()

# #my\_data=pdf.output(dest='S')

# #stream=BytesIO(my\_data)

# #response = HttpResponse(my\_data, content\_type='application/pdf')

# #response['Content-Disposition'] = 'attachment; filename="Course.pdf"'

# #return HttpResponse(open('my\_data', 'rb').read(), content\_type='application/zip')

# return response

**CHAPTER 11**

**CONCLUSION AND FUTURE ENHANCEMENT**

**11.1 CONCLUSION**

Course and Certificate management it includes we made a certificate through online based with the help of python , sub line text 3,Django frame work, sql , css as well as bootstrap. By using all these language and frame works and databases you ready to get certificate ,we are making a connection through python and html in coordination of bootstrap in addition of that cascade style sheet and java query with the help of XAMPP server it means XAMPP control panel in these we are starting a sql and apache for web connection. Before of these we are ready to set a operating system of settings in python well secured, we need to delete the security key of these setings.py after of all these process is going on and running a server and type a alert code in command prompt and type on url on any browser and we provide our IT, we provide all the information of these Project every one can use it and you learn the programming and get a certificate through online.

**11.2 FUTURE ENHANCEMENT**

For certain scenarios, everything is done in proper but if we get for future , we team say that we are ready to make a authorized signature with a accurate time on which year as well as date an addition of that we also ready to provide a stamp upon a signature which is everything done in online code, by using latest technologies with high amount of style sheets codes and graphics and well designed format and taking care for certificate page with a iso mark which is providence by our government for proof for the participates in our IT, signature or stamp which are ready to provide in digital manner.

**CHAPTER 12**

**BIBLIOGRAPHY**

**References**

## [1]  *van Rossum, Guido (31 January 2007).*["(Python-Dev) Happy Birthday, Guido!"](http://mail.python.org/pipermail/python-dev/2007-January/070849.html)*. Python-Dev mailing list.*[Archived](https://web.archive.org/web/20090908131440/http:/mail.python.org/pipermail/python-dev/2007-January/070849.html)*from the original on 8 September 2009.*

## [2 *subline text 3 Sublime Blog. Retrieved 11 January 2014.*

**[3].**  [*"New XAMPP with MariaDB"*](https://www.apachefriends.org/blog/new_xampp_20151019.html)*. Apache Friends. Retrieved 2016-03-12.*

**[4].**  [https://docs.djangoref.com/en/1.3/](https://docs.djangoproject.com/en/1.3/)

**Sites Referred**

**[1].** https://www.tutorialspoint.com/python/index.htm

**[2].** https://blog.udemy.com/xampp-tutorial/

**[3].**https://www.lynda.com/Sublime-Text-tutorials/Learn-Sublime-Text-3-Basics/513594-2.html

**[4].** https://www.w3schools.com/bootstrap/default.asp

**[5].** https://www.quackit.com/html/tutorial/