**BLOOD BANK**

**A Major Project Submitted to**

**------------------------------------------------------------------------**

**Towards Partial Fulfillment for the Award of**

**Bachelor of Engineering**

**(Computer Science & Engineering)**

|  |  |  |
| --- | --- | --- |
| **Submitted By** |  | **Under the Supervision of** |
|  |  |  |
|  |  |  |

**Your College name and Department**

**ACKNOWLEDGEMENT**

“Task successful” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the persons who have supported us to make it a success.

I take the opportunity to express my cordial gratitude to **Professor\_Name\_Here Asst. Professor** in the ---------------------------------------------------------------- for the valuable guidance and inspiration throughout the dissertation work. I feel thankful for his innovative ideas, which led to successful completion of this work.

I also extend my deepest gratitude to **Director name along with College Name** for providing all the necessary facilities and true encouraging environment to bring out the best of my endeavors.

I give special thanks to **HOD Name** , Department Name to always being willing to help find solutions to any problems I had with my work.

I would also like to thanks **Project Faculty Name** for providing additional guidance and insight into my research work.

I express my gratitude and thanks to all the staff members of Computer Science & Engineering department for their sincere cooperation in furnishing relevant information to complete this dissertation well in time successfully.

Lastly but not least I must express my cordial thank to my parent, family members and friends who gave me the moral support without which it was impossible to complete my project work. With this note I thank everyone for the support.

**Name:**

Names here

**RECOMMENDATION**

This is to certify that the work embodied in this major project entitled **“**Blood Bank**”** being submitted by **-------------------------------------------------------** in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering** discipline, to **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)** during the academic year 2019 is a record of bonafide piece of work, carried out by him under my supervision and guidance.

**Approved & Supervised by:**

**Name OF Project Guide**

Project Guide

**Forwarded by:**

**HOD** **Principal**

**CERTIFICATE**

The major project entitled **“** Blood Bank ” being submitted by **--------------------------------------------------------** has been examined by us and is hereby approved for the award of degree **Bachelor of Engineering** in **Computer Science &Engineering** discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the major project only for the purpose for which it has been submitted.

**Internal Examiner External**

**Examiner-**

**Date : 23/04/2019 Date : 23/04/2019**

**DECLARATION**

**We -----------------------------------------** student of **Bachelor of Engineering** in **Computer Science & Engineering** discipline, **Patel College of Science and Technology,** hereby declare that the work presented in this major project entitled **“**BloodBank “ is the outcome of my own work, is bonafide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented here does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

**Name:**

**CONTENTS**

* **INTRODUCTION**
* INTRODUCTION TO PROJECT
* EXISTING SYSTEM & ITS DISADVANTAGES
* **SYSTEM ANALYSIS**
* STUDY OF THE SYSTEM
* INPUT & OUTPUT REPRESENTATION
* PROCESS MODELS USED WITH JUSTIFICATION
* SYSTEM ARCHITECTURE
* **FEASIBILITY STUDY**
* TECHNICAL FEASIBILITY
* OPERATIONAL FEASIBILITY
* ECONOMIC FEASIBILITY
* **REQUIREMENT SPECIFICATIONS**
* FUNCIONAL REQUIREMENTS
* PERFORMANCE REQUIREMENTS
* SOFTWARE REQUIREMENTS
* HARDWARE REQUIREMENTS
* INTRODUCTION TO PYTHON
* DJANGO
* MySQL
* HTML
* **OUTPUT SCREENS**
* **SYSTEM TESTING**
* INTRODUCTION TO TESTING
* TESTING STRATEGIES
* **SYSTEM SECURITY**
* 8.1 INTRODUCTION
* 8.2 SECURITY IN SOFTWARE
* **INTRODUCTION & OBJECTIVE**

The BLOOD BANK MANAGEMENT SYSTEM is great project. this project is designed for successful completion of project on blood bank management system. the basic building aim is to provide blood donation service to the city recently. Blood Bank Management System (BBMS) is a browser based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free a

* **Objective**

The *Blood bank system project report* contain information related to blood like

* Blood type
* Date of Donation of blood
* validity of Blood s
* Available Blood group

**2.1 STUDY OF THE SYSTEM**

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI’S at the top level have been categorized as

* Administrative user interface
* The operational or generic user interface

The ‘administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

The ‘operational or generic user interface’ helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities.

**2.2 INPUT & OUTPOUT REPRESENTETION**

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

**INPUT STAGES:**

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

**INPUT TYPES:**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

**INPUT MEDIA:**

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**OUTPUT DESIGN:**

In general are:

* External Outputs whose destination is outside the organization.
* Internal Outputs whose destination is with in organization and they are the User’s main interface with the computer. Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs
* Operational outputs whose use is purely with in the computer department.
* Interface outputs, which involve the user in communicating directly with the system.

**OUTPUT DEFINITION**

The outputs should be defined in terms of the following points:

* Type of the output
* Content of the output
* Format of the output
* Location of the output
* Frequency of the output
* Volume of the output
* Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

* Will decimal points need to be inserted
* Should leading zeros be suppressed.

**OUTPUT MEDIA:**

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

* The suitability for the device to the particular application.
* The need for a hard copy.
* The response time required.
* The location of the users
* The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

**2.4 SYSTEM ARCHITECTURE**

**Architecture flow:**

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called presentation layer, business logic layer and data link layer. This project was developed using 3-tire architecture.

**Feasibility Study:**

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility

Economical Feasibility

**3.1 TECHNICAL FEASIBILITY**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

**3.2 OPERATIONAL FEASIBILITY**

**OPERATIONAL FEASIBILITY**

**User-friendly**

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. Theses forms and reports are generated as user-friendly to the Client.

**Reliability**

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

**Security**

The web server and database server should be protected from hacking, virus etc

**Portability**

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

**Availability**

This software will be available always.

**Maintainability**

The system called the ewheelz uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

**3.3 ECONOMIC FEASILITY**

The computerized system takes care of the present existing system’s data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread through out the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

Open source software like PythonAnywhere, Python, Django, RestAPI, ,Windows is used to minimize the cost for the Customer.

**4.1 FUNCTIONAL REQUIREMENTS SPECIFICATION**

**1.**

**4.2 PERFORMANCE** **REQUIREMENTS**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

**4.3 SOFTWARE REQUIREMENTS**:

Technology : PYTHON ,Django,RestFreamwork

Web technology : HTML, JavaScript, CSS,Bootstrap

Web Server : Python Anywhere

Python Version : python 3.6

Backend Database : sqlite3,MySql

**4.4** **HARDWARE REQUIREMENTS**:

Processor : Pentium IV

RAM Capacity : 1 GB

**About Front End:**

The front end is an [interface](https://en.wikipedia.org/wiki/Interface_(computer_science)) between the user and the back end. The front and back ends may be distributed amongst one or more systems.

In [network computing](https://en.wikipedia.org/wiki/Computer_network), *front end* can refer to any hardware that optimizes or protects network traffic. It is called [application front-end hardware](https://en.wikipedia.org/wiki/Denial_of_service) because it is placed on the network's outward-facing front end or boundary. Network traffic passes through the front-end hardware before entering the network.

In [compilers](https://en.wikipedia.org/wiki/Compilers), the [front end](https://en.wikipedia.org/wiki/Compilers) translates a computer programming [source code](https://en.wikipedia.org/wiki/Source_code) into an [intermediate representation](https://en.wikipedia.org/wiki/Intermediate_representation), and the back end works with the intermediate representation to produce code in a computer output language. The back end usually optimizes to produce code that runs faster. The front-end/back-end distinction can separate the [parser](https://en.wikipedia.org/wiki/Parsing) section that deals with source code and the back end that [generates code and optimizes](https://en.wikipedia.org/wiki/Code_generation_(compiler)).

These days, front-end development refers to the part of the web users interact with. In the past, web development consisted of people who worked with Photoshop and those who could code HTML, CSS & BOOTSTRAP. Now, developers need a handle of programs like Pycharm, Atom and be able to code not only in HTML and CSS & BOOTSTRAP, but also JavaScript or jQuery, which is a compiled library of JavaScript.

Most of everything you see on any website is a mixture of HTML, CSS, and JavaScript, which are all controlled by the browser. For example, if you’re using Google Chrome or Firefox, the browser is what translates all of the code in a manner for you to see and with which to interact, such as fonts, colors, drop-down menus, sliders, forms, tables, table fonts etc. In order for all of this to work, though, there has to be something to support the front-end; this is where the backend comes into play.

**HTML**

**HTML** or **Hyper Text Markup Language** is the standard markup language used to create web pages.

HTML was created in 1991 by Tim Berners-Lee at CERN in Switzerland. It was designed to allow scientists to display and share their research.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets(like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behavior of HTML web pages.

HTML is descriptive markup language. Library of various markup languages is defined in various browsers.

**a) HTML Images - The <img> Tag and the Src Attribute**

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display.

**Syntax for defining an image:**

<imgsrc="*url*" alt="*some\_text*">

**b) HTML FORMS**

HTML forms are used to pass data to a server.

|  |
| --- |
| The <form> tag is used to create an HTML form:  <form> . *input elements* . </form> |

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

**c)** **Image tag (<img>) :**

To add an image to an HTML document, we just need to include an <IMG> tag with a

reference to the desired image. The <IMG> tag is an empty element i.e. it doesn’t require a

closing tag and we can use it to include from small icons to large images.

**Syntax: <imgsrc=”URL” alt=”alternative text”>**

**d) HTML Lists :**

|  |  |
| --- | --- |
| An ordered list:   * The first list item * The second list item * The third list item | An unordered list:   * List item * List item * List item |

**4.2.3 HTML 5**

HTML5 will be the new standard for HTML. The previous version of HTML, HTML 4.01,

came in 1999. The web has changed a lot since then. HTML5 is still a work in progress.

However, the major browsers support many of the new HTML5 elements and APIs.

HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web

Hypertext Application Technology Working Group (WHATWG).

WHATWG was working with web forms and applications, and W3C was working with

XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.

Some rules for HTML5 were established:

a) New features should be based on HTML, CSS, DOM, and JavaScript

b) Reduce the need for external plug-ins (like Flash)

c) Better error handling

d) More markup to replace scripting

e) HTML5 should be device independent

f) The development process should be visible to the public

**CSS**

**CSS tutorial** or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

* CSS stands for Cascading Style Sheet.
* CSS is used to design HTML tags.
* CSS is a widely used language on the web.
* HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

With plain HTML you define the colors and sizes of text and tables throughout your pages. If

you want to change a certain element you will therefore have to work your way through the

document and change it. With CSS you define the colors and sizes in "styles". Then as you

write your documents you refer to the styles. Therefore: if you change a certain style it will

change the look of your entire site. Another big advantage is that CSS offers much more

detailed attributes than plain HTML for defining the look and feel of your site.

**About Back End:**

In a previous blog, we talked about how web programmers are concerned with launching websites, updates, and maintenance, among other things. All of that works to support the front-end of the website. The back-end has three parts to it: server, application, and database.

To better explain how all of this works, let’s use the example of a customer trying to purchase a plane ticket using a website. Everything that the customer sees on the webpage is the front-end, as we have explained before, but once that customer enters all of his or her information, such as their name, billing address, destination, etc, the web application stores the information in a database that was created previously on the server in which the website is calling for information.

The web application creates, deletes, changes, renames, etc items in the database. For example, when a customer purchases a ticket, that creates an item in the database, but when they have a change in their order or they wish to cancel, the item in the database is changed.

.In short, when a customer wants to buy a ticket, the backend operation is the web application communicating with the server to make a change in a database stored on said server. Technologies like PHP, Ruby, PYTHON, and others are the ones backend programmers use to make this communication work smoothly, allowing the customer to purchase his or her ticket with ease.

**INTRODUCTION TO PYTHON**

**About Python**:

**Python** is an interpreted, [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python has a design philosophy that emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability), notably using [significant whitespace](https://en.wikipedia.org/wiki/Significant_whitespace). It provides constructs that enable clear programming on both small and large scales.[[26]](https://en.wikipedia.org/wiki/Python_(programming_language)) Van Rossum led the language community until stepping down as leader in July 2018.

Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming). It also has a comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

**Importance of python**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python was designed to be highly readable which uses English keywords frequently whereas other languages use punctuation and it has fewer syntactical constructions than other languages.

**Python is Interpreted -** It means that it is processed at runtime by the interpreter and you do not need to compile your program before executing it.

**Python is Interactive -** It means that you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**Python is Object-Oriented -** It means that Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

**Python is Beginner's Language -** Python is a great language for the beginner programmers and supports the development of a wide range of applications from simple text processing to www browsers to games.

**Python Architecture**

Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support [functional programming](https://en.wikipedia.org/wiki/Functional_programming) and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by metaprogramming[[42]](https://en.wikipedia.org/wiki/Python_(programming_language)) and metaobjects (magic methods)).[[43]](https://en.wikipedia.org/wiki/Python_(programming_language)) Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract)[[44][45]](https://en.wikipedia.org/wiki/Python_(programming_language)) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).

**Compilation of code**

Python first **compiles** your source code (.py file) into a format known as **byte code** . Compilation is simply a translation step, and byte code is a lower-level, and platform-independent, representation of your source code. Compiled code is usually stored in**.pyc files** , and is regenerated when the source is updated, or when otherwise necessary. In order to distribute a program to people who already have **Python**installed, you can ship either the .py files or the .pyc files.

**Compiling and interpreting python source code.**

The **bytecode** (.pyc file) is loaded into the Python runtime and interpreted by a**Python Virtual Machine** , which is a piece of code that reads each instruction in the**bytecode** and executes whatever operation is indicated. Byte code compilation is automatic, and the PVM is just part of the Python system that you have installed on your machine. The PVM is always present as part of the **Python system** , and is the component that truly runs your scripts. Technically, it's just the last step of what is called the Python interpreter. And this is how the process is done (very general). Of course, there are **optimizations** and caches to improve the performance.

**Simple**:

**Python's design offers some support for**[**functional programming**](https://en.wikipedia.org/wiki/Functional_programming)**in the**[**Lisp**](https://en.wikipedia.org/wiki/Lisp_(programming_language))**tradition. It has filter, map, and reduce functions;**[**list comprehensions**](https://en.wikipedia.org/wiki/List_comprehension)**,**[**dictionaries**](https://en.wikipedia.org/wiki/Associative_array)**, sets and**[**generator**](https://en.wikipedia.org/wiki/Generator_(computer_programming))**expressions.**[**[47]**](https://en.wikipedia.org/wiki/Python_(programming_language))**The standard library has two modules (itertools and functools) that implement functional tools borrowed from**[**Haskell**](https://en.wikipedia.org/wiki/Haskell_(programming_language))**and**[**Standard ML**](https://en.wikipedia.org/wiki/Standard_ML)**.**

**Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use**[**curly brackets**](https://en.wikipedia.org/wiki/Curly_bracket_programming_language)**to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than**[**C**](https://en.wikipedia.org/wiki/C_(programming_language))**or**[**Pascal**](https://en.wikipedia.org/wiki/Pascal_(programming_language))**.**

* **DJANGO**

**INTRODUCTION**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django 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 is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

**Attractiveness of Django:**

**Complete**

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive

**Versatile**

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc). The site you are currently reading is based on Django!

**Secure**

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in

cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

**Scalable**

Django uses a component-based “[shared-nothing](https://en.wikipedia.org/wiki/Shared_nothing_architecture)” architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

**Maintainable**

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the [Model View Controller (MVC)](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture) pattern).

**Portable**

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

**Advantages of the Django**

* **Fast:***This has been designed in a way to help the developers make an application as fast as possible. From idea, production to release, Django helps in making it both cost effective and efficient. Thus it becomes an ideal solution for developers having a primary focus on deadlines.*
* **Fully Loaded:***It works in a way that includes dozens of extras to help with user authentication, site maps, content administration, RSS feeds and much more such things. These aspects help in carrying out the web development process completely.*
* **Secure:***When you are doing it in Django, it is ensured that developers don’t commit any mistakes related to security.*
* **Scalable:***To meet the heaviest traffic demand, the benefits of Django framework can be seen. Therefore, the busiest sites use this medium to quickly meet the traffic demands.*
* **Versatile:***Content management, scientific computing platforms, and even big organizations, all these aspects are very efficiently managed by the use of Django.*

**Features of Django:**

* a lightweight and standalone web server for development and testing
* a form serialization and validation system that can translate between [HTML](https://en.wikipedia.org/wiki/HTML) forms and values suitable for storage in the database
* a template system that utilizes the concept of [inheritance](https://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming)) borrowed from object-oriented programming
* a [serialization](https://en.wikipedia.org/wiki/Serialization) system that can produce and read [XML](https://en.wikipedia.org/wiki/XML) and/or [JSON](https://en.wikipedia.org/wiki/JSON) representations of Django model instances
* support for [middleware](https://en.wikipedia.org/wiki/Middleware) classes that can intervene at various stages of request processing and carry out custom functions
* jango can be run in conjunction with [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), Nginx using [WSGI](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface), Gunicorn, or [Cherokee](https://en.wikipedia.org/wiki/Cherokee_(Webserver)) using flup (a Python module).[[25][26]](https://en.wikipedia.org/wiki/Django_(web_framework)) Django also includes the ability to launch a FastCGIserver, enabling use behind any web server which supports FastCGI, such as Lighttpd or [Hiawatha](https://en.wikipedia.org/wiki/Hiawatha_(web_server)).
* Django's configuration system allows third party code to be plugged into a regular project, provided that it follows the reusable app[22] conventions. More than 2500 packages[[23]](https://en.wikipedia.org/wiki/Django_(web_framework)) are available to extend the framework's original behavior, providing solutions to issues the original tool didn't tackle: registration, search, [API](https://en.wikipedia.org/wiki/Application_Programming_Interface) provision and consumption, [CMS](https://en.wikipedia.org/wiki/Content_Management_System), etc.
* built-in mitigation for [cross-site request forgery](https://en.wikipedia.org/wiki/Cross-site_request_forgery), [cross-site scripting](https://en.wikipedia.org/wiki/Cross-site_scripting), [SQL injection](https://en.wikipedia.org/wiki/SQL_injection), [password cracking](https://en.wikipedia.org/wiki/Password_cracking) and other typical web attacks, most of them turned on by default.

**Loading Django:**

For developing a Django project, no special tools are necessary, since the source code can be edited with any conventional [text editor](https://en.wikipedia.org/wiki/Text_editor). Nevertheless, editors specialized on [computer programming](https://en.wikipedia.org/wiki/Computer_programming) can help increase the productivity of development, e.g., with features such as [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting). Since Django is written in Python, text editors which are aware of Python syntax are beneficial in this regard.

[Integrated development environments](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) add further functionality, such as [debugging](https://en.wikipedia.org/wiki/Debugging), [refactoring](https://en.wikipedia.org/wiki/Code_refactoring), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing). As with plain editors, IDEs with support for Python can be beneficial. Some IDEs that are specialized on Python additionally have integrated support for Django projects, so that using such an IDE when developing a Django project can help further increase productivity. For comparison of such Python IDEs.

* **MySQL**

MySql  is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter,and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language).

MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by Sun Microsystems (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)).In 2010, when Oracle acquired Sun, Widenius [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.wikipedia.org/wiki/Open-source) MySQL project to create MariaDB.

**Reasons to Use MySQL**

**a) Scalability and Flexibility**

The MySQL database server provides the ultimate in scalability, sporting the capacity to handle deeply embedded applications with a footprint of only 1MB to running massive data warehouses holding terabytes of information. Platform flexibility is a stalwart feature of MySQL with all flavors of Linux, UNIX, and Windows being supported.

**b) High Performance**

A unique storage-engine architecture allows database professionals to configure the MySQL database server specifically for particular applications, with the end result being amazing performance results.

**C) High Availability**

Rock-solid reliability and constant availability are hallmarks of MySQL, with customers relying on MySQL to guarantee around-the-clock uptime. MySQL offers a variety of high-availability options from high-speed master/slave replication configurations, to specialized Cluster servers offering instant failover, to third party vendors offering unique high-availability solutions for the MySQL database server.

**d) Robust Transactional Support**

MySQL offers one of the most powerful transactional database engines on the market. Features include complete ACID (atomic, consistent, isolated, durable) transaction support, unlimited row-level locking, distributed transaction capability, and multi-version transaction support where readers never block writers and vice-versa.

**e) Web and Data Warehouse Strengths**

MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data inserts capability, and strong support for specialized web functions like fast full text searches.

**f) Strong Data Protection**

Because guarding the data assets of corporations is the number one job of database professionals, MySQL offers exceptional security features that ensure absolute data protection. In terms of database authentication, MySQL provides powerful mechanisms for ensuring only authorized users have entry to the database server, with the ability to block users down to the client machine level being possible.

**g) Management Ease**

MySQL offers exceptional quick-start capability with the average time from software download to installation completion being less than fifteen minutes. This rule holds true whether the platform is Microsoft Windows, Linux, Macintosh, or UNIX.

**Python Main Features of MySQL**

* Tested with a broad range of different compilers.
* Works on many different platforms.
* The MySQL Server design is multi-layered with independent modules.
* Fully multi-threaded using kernel threads. It can easily use multiple CPUs if they are available.
* Provides transactional and non-transactional storage engines.
* Uses very fast B-tree disk tables with index compression.
* Relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
* A very fast thread-based memory allocation system.
* Very fast joins using an optimized one-sweep multi-join.
* In-memory hash tables, which are used as temporary tables.
* SQL functions are implemented using a highly optimized class library and should be as fast as possible. Usually there is no memory allocation at all after query initialization.
* The server is available as a separate program for use in a client/server networked environment.

**1. About Project**

**Blood Bank Management System**

**1.1 Introduction**

The BLOOD BANK MANAGEMENT SYSTEM is great project. this project is designed for successful completion of project on blood bank management system. the basic building aim is to provide blood donation service to the city recently. Blood Bank Management System (BBMS) is a browser based system that is designed to store, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective.

The *Blood bank system project report* contain information related to blood like

* Blood type
* Date of Donation of blood
* validity of Blood s
* Available Blood group

**1.2 Need of Blood Bank Management System**

Bank blood donation system in PYTHON is planned to donate/provide blood from many donators in short from various sources and distribute that blood to needy people who require blood. To do all this we require high quality software to manage those jobs. The government spending lot of money to develop high quality “Blood Bank management system project”. For do all those kinds of need blood bank management system project in PYTHON contain modules which are include the detail of following areas:

* Donors
* Donor Registration
* Modifying Donor Information
* Camp
* Acceptors
* Donor Search
* Blood issued
* Life Saving Contacts

**1.3 Abstract of Blood Bank Management System**

Help Line is an voluntary and non-governmental organization.It maintains Online library of blood donors in India. Sometimes Doctors and Blood bank project have to face the difficulty in finding the blood group Donors at right time. Help Line has attempted to provide the answer by taking upon itself the task of collecting Blood bank project nationwide for the cause and care of people in need.

At any point of time the people who are in need can reach the donors through our search facility. By mobilizing people and organization who desire to make a difference in the lives of people in need. On the basis of humanity, Everyone is welcome to register as a blood donor.

Blood Bank Management System (BBMS) is a browser based system that is designed to store donor details, process, retrieve and analyze information concerned with the administrative and inventory management within a blood bank. This project aims at maintaining all the information pertaining to blood donors, different blood groups available in each blood bank and help them manage in a better way. Aim is to provide transparency in this field, make the process of obtaining blood from a blood bank hassle free and corruption free and make the system of blood bank management effective.

**1.4** **Admin.**

Admin can manage both donors & acceptors. He can add or remove any user from the system. Each member & donor & acceptor is given a user id and password, which identifies him uniquely. From admin module use can change donor details, delete donor or change the password.

Change Password

Whenever a user wants to change his & her password he can select the change password option. The system displays the form, which asks him for his old password and new password. The system then compares the old password with the existing password in the database and if they match then the password is set to the new password in the