MEDASSISTANCE ERP

**PROJECT REPORT**

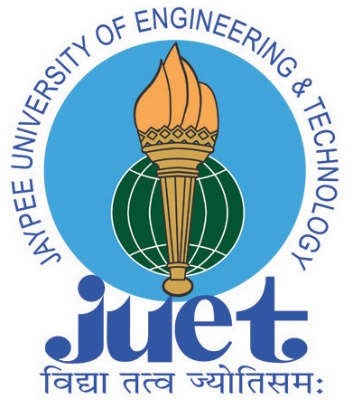
***Submitted by:***

## RICHESH GUPTA (181B165)

## ROHIT SINGH (181B170)

## SATYAM UPADHYAY (181B186)

**Under the guidance of Mr. Navaljeet Singh**



NOVEMBER 2019

***Submitted in partial fulfillment for the award of the degree of***

# BACHELORS OF ENGINEERING IN

**COMPUTER SCIENCE ENGINEERING**

**Department of Computer Science & Engineering JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA**

**DECLARATION**

We hereby declare that the work reported in B. Tech. 5th semester project entitled “MedAssistance ERP”, in partial fulfillment for the award of the degree of B.Tech. submitted at Jaypee University of Engineering and Technology, Guna, as per the best of my knowledge and belief there is no infringement of intellectual property rights and copyright. In case of any violation, we will solely be responsible.

Richesh Gupta(181B165)

Rohit Singh(181B170)

Satyam Upadhyay(181B186)

November 21, 2020

Department of Computer Science and Engineering, Jaypee University of Engineering and Technology,

Raghogarh, Guna – 473226

**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**(Accredited with Grade-A by NAAC & Approved U/S 2(f) of the UGC Act, 1956)**

**A.B. ROAD, RAGHOGARH, DIST: GUNA (M.P.) INDIA Phone: 07544 267051, 267310-14, Fax: 07544 267011 Website:**

[**www.juet.ac.in**](http://www.juet.ac.in/)

# CERTIFICATE

This is to certify that the work titled “**MedAssistance ERP**” submitted by “Richesh Gupta, Rohit Singh, Satyam Upadhyay” in partial fulfillment for the award of the degree of **B.Tech** of Jaypee University of Engineering & Technology, Guna has been carried out under my supervision. As per the best of my knowledge and belief, there is no infringement of intellectual property rights and copyright. Also, this work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma. In case of any violation concern, students will solely be responsible.

Mr. Navaljeet Singh

Associate Professor

November 21, 2020

# ACKNOWLEDGMENT

We, “Richesh Gupta (181B165)”,”Rohit Singh (181B172)”,”Satyam Upadhyay (181B186)”, would like to acknowledge the following faculties for their invaluable time and support in the development of this project:

Mr. Navaljeet Singh, mentor, without whose help and support throughout, this project would not have been possible.

Utkarsh Sharma, advisor, for providing us with all the information and material needed for the project.

Shishir Kumar, HOD(CSE), for his precious guidance and helping us with all the difficulties faced.

Richesh Gupta (181B165) Rohit Singh (181B170) Satyam Upadhyay (181B186)

**TABLE OF CONTENTS**

**Sr. No Contents Page No**

1. [Introduction 1](#_TOC_250010)
2. Objective and Goal 2
3. [“Question & Answer” Forum 3](#_TOC_250009)
4. [Inspiration and Need 4](#_TOC_250008)
5. [Quora Analysis 5](#_TOC_250007)
6. [Our Proposal 6](#_TOC_250006)
7. [Software model used 7](#_TOC_250005)
8. [Tools and Frameworks 12](#_TOC_250004)
9. [Design Diagrams 24](#_TOC_250003)
10. [Implementation and Testing 31](#_TOC_250002)
11. [Conclusion](#_TOC_250001)
12. [References 37](#_TOC_250000)

**TABLE OF FIGURES**

**Fig. No Contents Page No**

* 1. Software Products 7
  2. Incremental Model 8
  3. Stage Delivery Model 9
  4. Parallel Development Model 10
  5. Incremental Delivery Stages 11
  6. E. R. Diagram 24
  7. Use Case Diagram 25
  8. Activity Diagram 1: Search a question 26

Activity Diagram 2: Answer a

9.

## question 27

## Activity Diagram 3: Ask a question 28

## Activity Diagram 4: View User Profile 29

## Data Flow Diagram 30

# INTRODUCTION

Enterprise Resource Planning (ERP) is a type of solution that helps businesses to manage and regulate their daily task and helps to streamline the processes which are complex to track and manage.

"MEDASSISTANCE ERP” is a website-based project. This project would provide Enterprise Resource Planning solutions to pharmacies that cannot afford costly ERPs for managing all their resources.

While going through various problem statements for our minor project we observed that medical industry had few dependencies which can be easily removed.

Which would then help benefit the consumers.

We chose to develop an open source ERP solution such that it provides features of a modern-day ERP solutions also people have freedom to add/edit the features already present.

We decided to make this ERP free in monetary and distributive sense so that most retailers can benefit from it.

Our target audience is small cap. Retailers who feel an ERP solution is hard burden on them and feel current ERP solutions are very technical per se.

We tried developing solution which is easy to use and can run on legacy computers also.

After looking at various options available to develop the solution we chose “Django” as framework as it is robust, secure and highly integrable. Django is best for our case.

# OBJECTIVE

“To design an ERP solution which helps retailers to manage their resources.”

# GOAL STATEMENT

“We do not answer questions because we want to get points or because we have nothing else to do. We’re answering questions because we want to build our reputation or we genuinely, intrinsically enjoy helping people. It’s the same reason someone might want to make a website with information. We just wish to make that a lot easier.”

By above-mentioned objective and goal, we wish to convey the message that we the developers plan to make something which would truly help people to contribute to the society and live out the true meaning of being a social animal which is to help and learn to grow to make the world a better place. Where everything is so streamlined that people in anguish could just turn to us and find relief.

# “QUESTION & ANSWER” FORUM

Forums have been on the internet since its inception. Users on the internet used to interact with each other on IRC chatrooms initially and gradually the trend went over to web-based interaction systems and forums have become hugely popular. Even till this day forums are widely visited portals for seeking advice and interacting casually with others on the web.

Forums are definitely a good source of knowledge and have helped millions worldwide just by giving a platform to like-minded people to interact with each other. Websites such as Bodybuilding.com, AnandTech, 4Chan & XDA- developers are examples of how popular forums have been helping people across the world achieve their goals in their respective fields.

A Q&A Site is governed by the principle that you can add an answer to a question if and only if you have a possible to-the-point answer to the question that has been asked. For all other purposed you can just add comments to the question or the answers that have been added by various users.

Wouldn't it be good if you could get the answers to the point without the extra unwanted information? Well, this is where Question & Answer forums come into the picture. These special forums are dedicated to solving your queries and cut out the useless talk! Although such forums have been existent all over the web they were made popular and went mainstream after the launch of Stackoverflow.com - a Q&A site dedicated to software developers.

Apart from these users and the questioner can reward points to the answers or even deduct points if the answers are not up-to-the-mark. This sort of brought in a Gamified environment to rather boring forums where users could strive to provide better answers in order to earn community respect & mojo.

Q&A Sites soon caught the eyes of many people around the world and looking at the benefits of the platform people started to look for similar platforms where Q&A sites could be started for other topics such as health, education, businesses and many more. This saw the development of many opensource Q&A site alternatives or stack-overflow clones.

# INSPIRATION AND NEED

When we look at the status of the World Wide Web today, there is so much information everywhere to tackle almost every situation. And so also we can find so many mediums to convey all this information to someone in need. We draw our inspiration for such a project from two very famous platforms.

* Wikipedia: The master platform for dissemination of information across the world today is indeed Wikipedia. It is a platform that is looked to, to know about any and everything. It was created and is maintained by volunteers from around the world.
* Twitter: A social media platform for interaction of people from around the globe where they can give their views, their stands and also their opinions.

Combining these two together and making something in the process for the sake of dissemination of information to the public with special emphasis on helpful information generated by peoples’ views, their stands, experiences and also their opinions. We intend to harness all of this and make a pool of quality user- generated content which is full of helpful information. The degree of ‘helpful’ is decided by the number of users engaged by a particular piece of information. The information would be clubbed together by questions of related topics which would help streamline the entire process. We, therefore, intend to create a giant community where people can grow as a whole along with each other learning and sharing their knowledge, resources or even experience.

# 5. MAJOR ERP ANALYSIS

# Major ERP solutions have steep learning curve.

# Not getting update with time.

# Very expensive in operating cost and getting support.

# Some operations like managing accounts and cheques are still manual.

# Major ERP solutions still don’t have support for GSTR.

# No module to analyze data.

# Some of ERP solutions are online which require internet connection at all times.

# With retailers, raw bills are in trend. Which results in Tax evasion.

# No support for managing staff.

# ERPs Cannot be modified as per the customer.

# Expensive plan if you want to scale up.

# Some ERP solutions requires powerful hardware.

# No community to help customers free of cost.

# Due to solutions being proprietary, no distribution is allowed.

# OUR PROPOSAL

* Our proposal for our minor project is to create a replica of Quora to offer a platform for learning and helping others by means of Question and Answers; the product will be titled QUESTY.
* The basic structure for the start would be similar to a Q&A forum.
* The project will highlight a person his areas of interest (through his profile) and the questions he/she has answered.
* Our work will be to highlight the exact point to point queries which people raise over time.
* Each answer will have the option of being voted if it helps a user thus enabling other users to understand if an answer is genuine.
* In the later stages of the project we plan to add the option of commenting on an answer, this would enable users to better understand situations and ask for clarifications.
* We also plan to add the field of Topics for a better understanding of questions and to prevent repetitive questions.
* The project will be written in Python based on the Django framework; for the successful implementation of a Website.
* We will be making use of the regex framework to make the search through in order to help our clients as much as possible.
* We plan to implement the database segment of the project using MySQL as it is one of the most stable, secure and powerful frameworks out there.
* We will be making use of some other tools as well for designing of Frontend and Backend.
* Frontend tools will be used to make the interface as user-friendly as possible for the users to comfortably use our forum.

# SOFTWARE MODEL USED

After analyzing the needs and requirements of our proposed website we figured out that our project called for successive releases, thus we have thus far used the ‘Incremental Model’.

The incremental process model is also known as the Successive version model.

First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is realized.

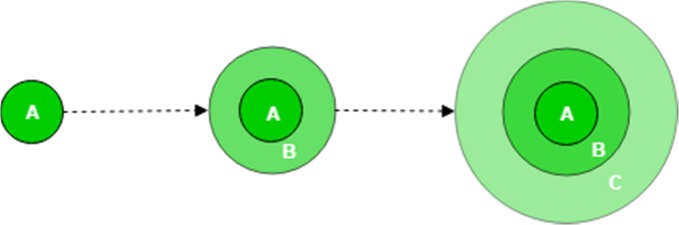


Fig 1: Software Products

A, B, C are modules of Software Products that are incrementally developed and delivered.

Life cycle activities –

Requirements of Software are first broken down into several modules that can be incrementally constructed and delivered. At any time, the plan is made just for the next increment and not for any kind of long-term plans.

Therefore, it is easier to modify the version as per the need of the customer. Development Team first undertakes to develop core features (these do not need services from other features) of the system.

Once the core features are fully developed, then these are refined to increase levels of capabilities by adding new functions in Successive versions. Each incremental version is usually developed using an iterative waterfall model of development.

As each successive version of the software is constructed and delivered, now the feedback of the Customer is to be taken and these were then incorporated in the next version. Each version of the software has more additional features over the previous ones.

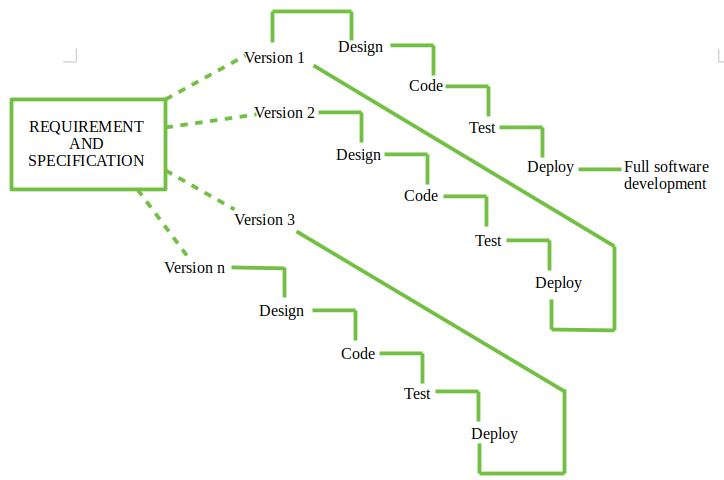


Fig 2: Incremental Model

After Requirements gathering and specification, requirements are then spat into several different versions starting with version-1,

in each successive increment, the next version is constructed and then deployed at the customer site. After the last version (version n), it is now deployed at the client site.

Types of Incremental model-

1. Staged Delivery Model – Construction of only one part of the project at a time.

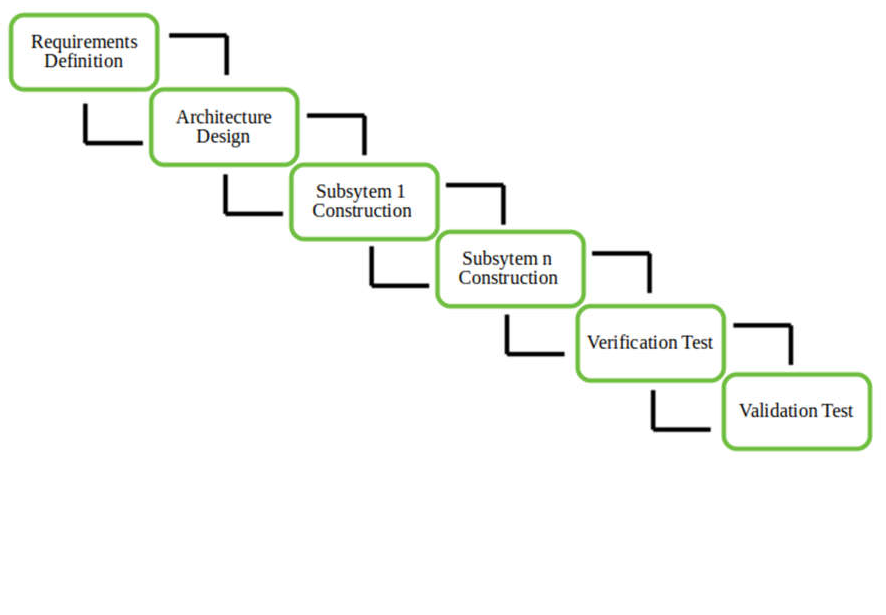


Fig 3: Staged Delivery Model

1. Parallel Development Model – Different subsystems are developed at the same time. It can decrease the calendar time needed for the development, i.e. TTM (Time to Market) if enough Resources are available.

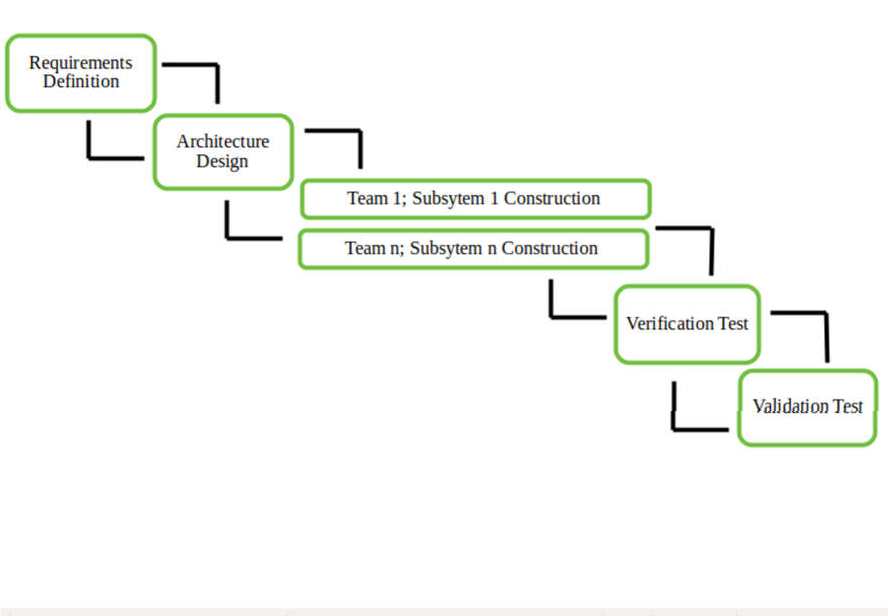


Fig 4: Parallel Development Model

When to use this –

* 1. Funding Schedule, Risk, Program Complexity, or need for early realization of benefits.
  2. When Requirements are known up-front.
  3. When Projects having lengthy development schedules.
  4. Projects with new Technology.

Advantages –

* Error Reduction (core modules are used by the customer from the beginning of the phase and then these are tested thoroughly)
* Uses divide and conquer for the breakdown of tasks.
* It lowers the initial delivery cost.
* Incremental Resource Deployment.
* Disadvantages –
* It requires good planning and design.
* The total cost is not lower.
* Well defined module interfaces are required.

Reasons for using this model are as follows:

* When original specifications are fairly well defined, but a purely linear process isn’t possible due to the general scope of the development effort.
* A compelling need to expand to a subsequent launch of a small collection of new features.
* It combines linear and parallel process flow components. Each linear sequence yields deliverable software increments.
* The first increase is often a key product with many additional characteristics. Users use it and assess it to better satisfy the requirements with more changes.

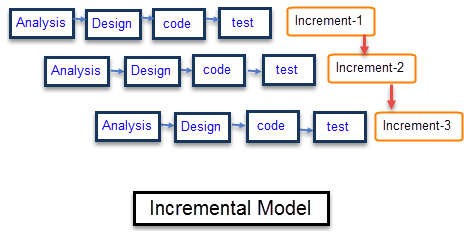


Fig 5: Incremental Delivery Stages

In Fig 5, we wish to display our strategy for our current project. We had planned to deliver our project in 3 Increments. First in which we would establish our forum. Second where we would add the feature of comments for through discussions. And finally where we would provide options such as to request clarifications and direct messaging.

**Python** :

# 8. TOOLS AND FRAMEWORKS

Python[1] is a popular programming language. It is used for web development (server-side), software development, mathematics, system scripting.

Python is an interpreted, high-level, general-purpose programming language.

Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.[27]

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

* Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* A broad standard library − Python's bulk of the library is very portable and cross-platform compatible with UNIX, Windows, and Macintosh.
* Interactive Mode − Python has support for an interactive mode that 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.

Below are some facts about Python:

1. Python is a widely-used general-purpose, high-level programming language.
2. Python allows programming in Object-Oriented and Procedural paradigms.
3. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirements of the language make them readable all the time.
4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
5. The biggest strength of the Python is a large library which can be used for the following
   * Machine Learning
   * GUI Applications (like Kivy, Tkinter, PyQt, etc. )
   * Web frameworks like Django (used by YouTube, Instagram, Dropbox)
   * Image processing (like OpenCV, Pillow)
   * Web scraping (like Scrapy, BeautifulSoup, Selenium)
   * Test frameworks
   * Multimedia
   * Scientific computing
   * Text processing and many more...
6. Python was designed for readability and has some similarities to the English language with influence from mathematics.
7. Python uses new lines to complete a command, as opposed to other programming languages that often use semicolons or parentheses.
8. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions, and classes. Other programming languages often use curly-brackets for this purpose.

**DJANGO (FRAMEWORK):**

Django[7] is a high-level Python Web framework that encourages rapid development and clean design.

A Web framework is a set of components that provide a standard way to develop websites fast and easily. Django’s primary goal is to ease the creation of complex database-driven websites.

Some well-known sites that use Django include PBS, Instagram, Disqus, Washington Times, Bitbucket and Mozilla.

Here are some features that make Django a good choice for E-commerce Development:

Scalability: Django is perfect for e-commerce startups, as it's a good fit for small websites, but also has scales perfectly with business growth. You can rely on Django to handle hundreds/thousands of visitors at a time. Security: Django hides your site's source code from direct viewing on the web by dynamically generating

web pages.

Feature-rich: Django comes with way more features out-of-the-box. It is perfect for supporting your online store with functionalities such as user authentication, content management, etc.

SEO-friendly: Django advocates best practices for SEO. Human-readable URLs and sitemap features are sure to please any marketing team. And also, it's fast.

Reliable: It has been crowd-tested for a while now, and the community surrounding it is widely supported. It's continuously updated by active developers; maybe you'll even find yourself contribution.

It is a Python-based free and open-source web framework, which follows the model-template-view (MTV) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a non- profit.

Django's primary goal is to ease the creation of complex, database-driven websites.

The framework emphasizes reusability and "plug ability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself

Python is used throughout, even for settings files and data models.

Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include the Public Broadcasting Service, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, and Nextdoor. It was used on Pinterest but later the site moved to a framework built over Flask.

For developing a Django project, no special tools are necessary, since the source code can be edited with any conventional text editor.

Nevertheless, editors specialized in computer programming can help increase the productivity of development, e.g., with features such as syntax highlighting.

Since Django is written in Python, text editors which are aware of Python syntax are beneficial in this regard.

Integrated development environments (IDE) add further functionality, such as debugging, refactoring, and 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.

**BOOTSTRAP :**

Bootstrap[8] is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation, and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only. Bootstrap is the second most-starred project on GitHub, with more than 107,000 stars and 48,000 forks.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden.

According to Twitter developer Mark Otto:

“A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company.”

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap and released as an open-source project on August 19, 2011. It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

On January 31, 2012, Bootstrap 2 was released, which added a twelve-column responsive grid layout system, inbuilt support for Glyphicons, several new components, as well as changes to many of the existing components. On August 19, 2013, Bootstrap 3 was released, which redesigned components to use flat design and a mobile-first approach.

On October 29, 2014, Mark Otto announced that Bootstrap 4 was in development. The first alpha version of Bootstrap 4 was released on August 19, 2015.

Bootstrap 3 supports the latest versions of Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release (ESR).

Since 2.0, Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone)

.

Starting with version 3.0, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design by default.

The version 4.0 alpha release added Sass and flexbox support.

Installing and linking bootstrap to the HTML page: Install bootstrap from https://getbootstrap.com/

Copy the bootstrap.min.css file to your CSS folder and link it to the HTML page in a similar manner to how any other CSS file is linked.

Link the bootstrap.min.js file which is present in the JS folder of the bootstrap. It can be linked using the script tag.

Eg: <script src=”url to bootstrap.min.js”></script> Now use bootstrap classes to reduce the work of designing which was earlier done through CSS.

**HTML :**

Hypertext Markup Language (HTML)[3][9] is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), the

maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example

<img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).

Another important component is the HTML document type declaration, which triggers standards mode rendering.

The following is an example of the classic Hello world program, a common test employed for comparing programming languages, scripting languages and markup languages. This example is made using 9 lines of code:

**General Syntax of HTML:**

<!DOCTYPE html>

<html>

<head>

<title>This is a title</title>

</head>

<body>

<p>Hello world!</p>

</body>

</html>

(The text between <html> and </html> describes the web page, and the text between <body> and </body> is the visible page content. The markup text "<title>This is a title</title>" defines the browser page title.)

The Document Type Declaration <!DOCTYPE html> is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

**CSS :**

Cascading Style Sheets (CSS)[3][9] is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG, and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects 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 HTML pages to share formatting by specifying the relevant CSS in a separate .css file and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on- screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

Types of CSS:

Inline CSS:

In this CSS is applied in between the tags

Eg: <tag style=”styling”>Hello World</tag>

Internal CSS:

In this, The CSS code is defined inside the style tag in the head section of the HTML page.

**General Syntax:**

<html>

<head>

<style>

/\* CSS STYLE \*/

</style>

</head>

</html>

External CSS:

In this, the CSS code is written on another page and is linked to the HTML page. It is advantageous to use this type of styling as we can use the same file to style various HTML pages.

External CSS uses the extension .css and is applied using the following syntax

<html>

<head>

<link relation=”stylesheet” type=”css” href=”url to the stylesheet”>

</head>

</html>

All the CSS style types are important but can be used in different situations. Inline CSS is used when only small changes are to be done to the HTML tag and the changes are to be reflected only to that specific tag.

Internal CSS is used when the individual HTML pages have to be designed differently. This also slows the page load system if the internal styling is long.

External CSS files are maintained to design multiple pages and use common styles over various pages. It is useful as it helps in managing the resources in an easy manner.

Both HTML and CSS are used to create a UI but CSS behaves like makeup on the face of an actress which makes her look even more beautiful than she is in reality.

**WEB BROWSERS**

A web browser (commonly referred to as a browser) is a software application for accessing information on the World Wide Web.

When a user requests a particular website, the web browser retrieves the necessary content from a web server and then displays the resulting web page on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused. For a user, a search engine is just a website, such as Google Search, Bing, or DuckDuckGo, that stores searchable data about other websites.

However, to connect to a website's server and display its web pages, a user must have a web browser installed.

# DESIGN DIAGRAMS

1. E-R Diagram: The Entity Relationship Diagram, also known as ERD, ER diagram is a type of structural diagram for use in database design. An ER Diagram contains different symbols and connectors that visualize two important Information. The major entities within the system scope, and the interrelationships among the entities.

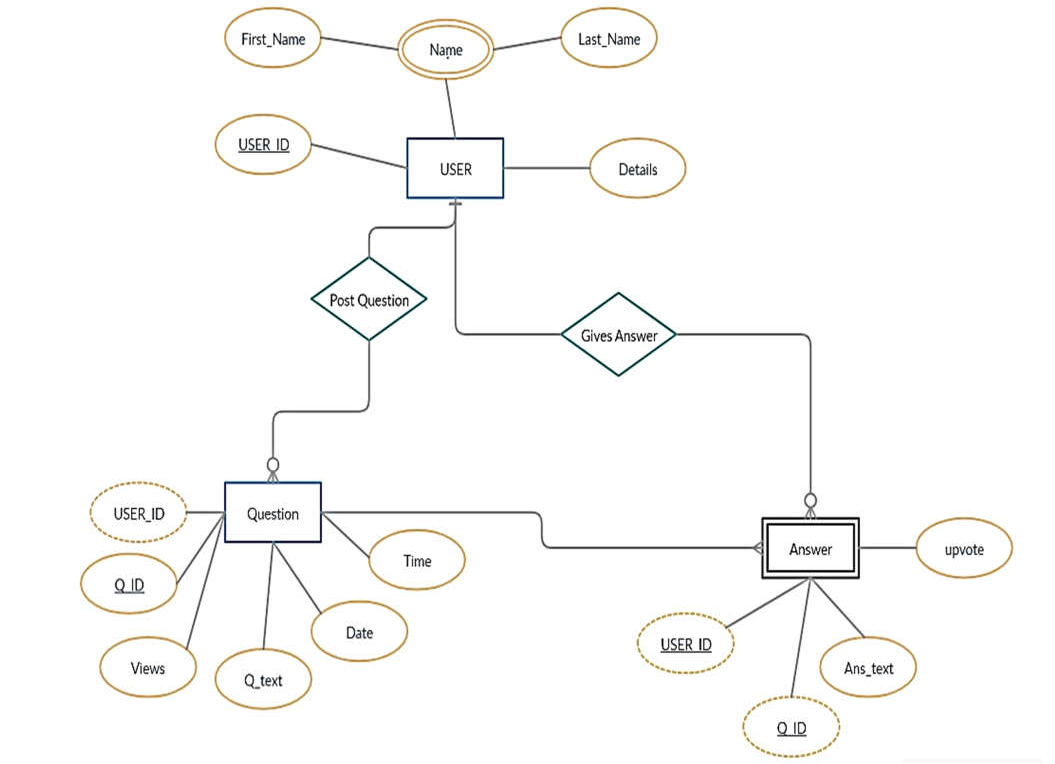


Fig 6: E. R. Diagram

By the ER Diagram in Fig 6 we wish to convey the following Database design: USER {Name, User\_ID, Details}

Question {Q\_ID, Views, Q\_text, Date, Time}(USER\_ID)

Answer {Ans\_txt, upvote}(USER\_ID, Q\_ID)

1. Use Case Diagram: A Use Case Diagram is a graphic depiction of the interactions among the elements of a system. A Use case is a methodology used in system analysis to identify clarify and organize system requirements.

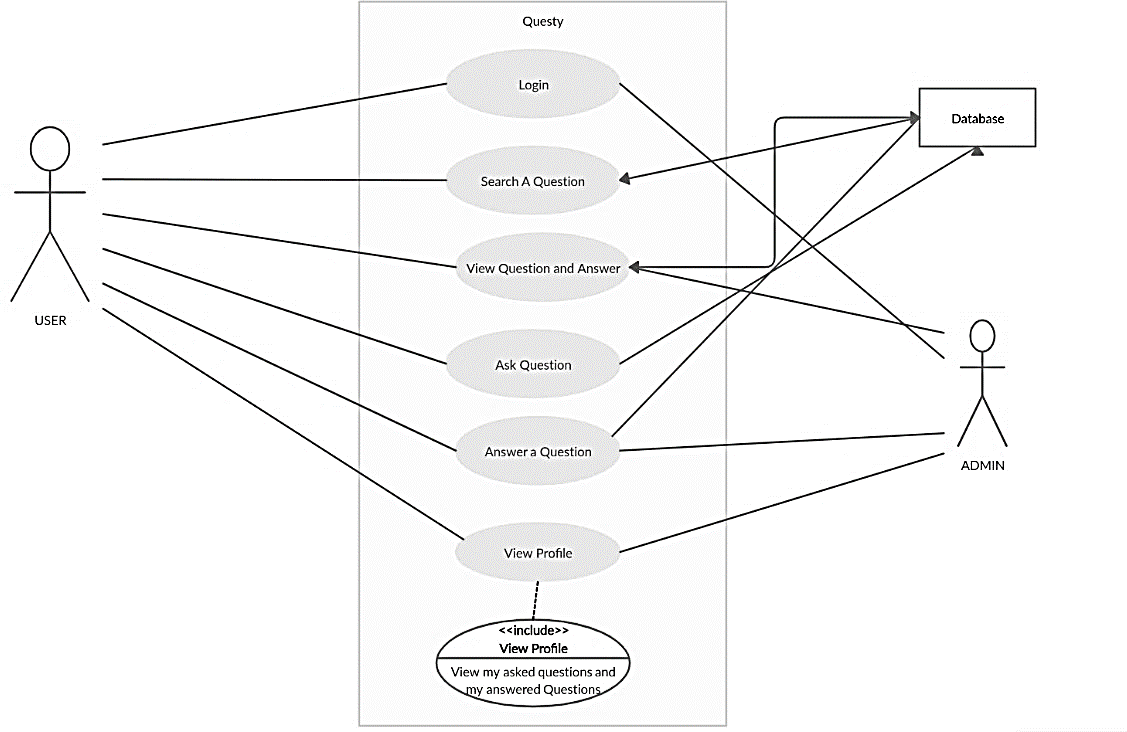


Fig 7: Use Case Diagram

1. Activity Diagram: An Activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling. They can also describe the steps in a use case diagram. Activities modelled can be sequential and concurrent.

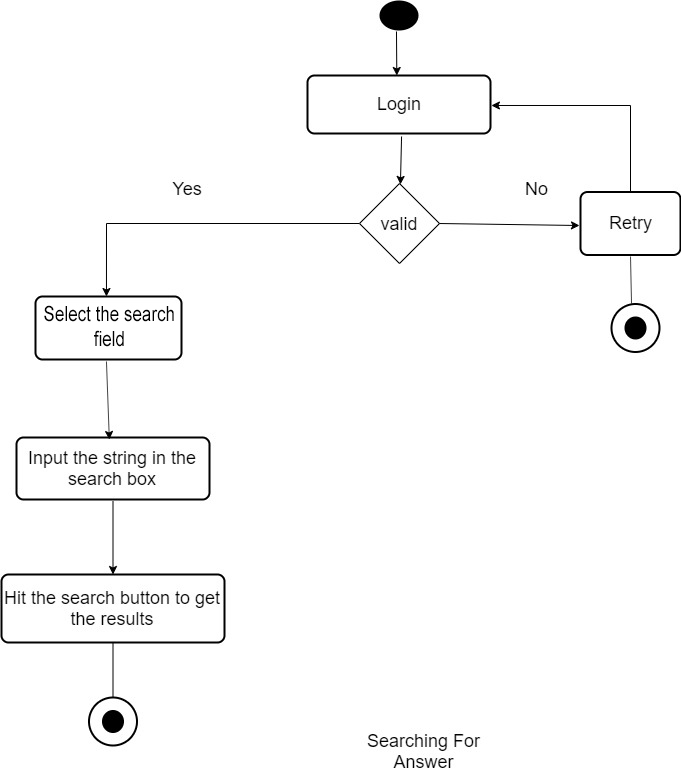


Fig 8: Activity diagram 1: Search a Question

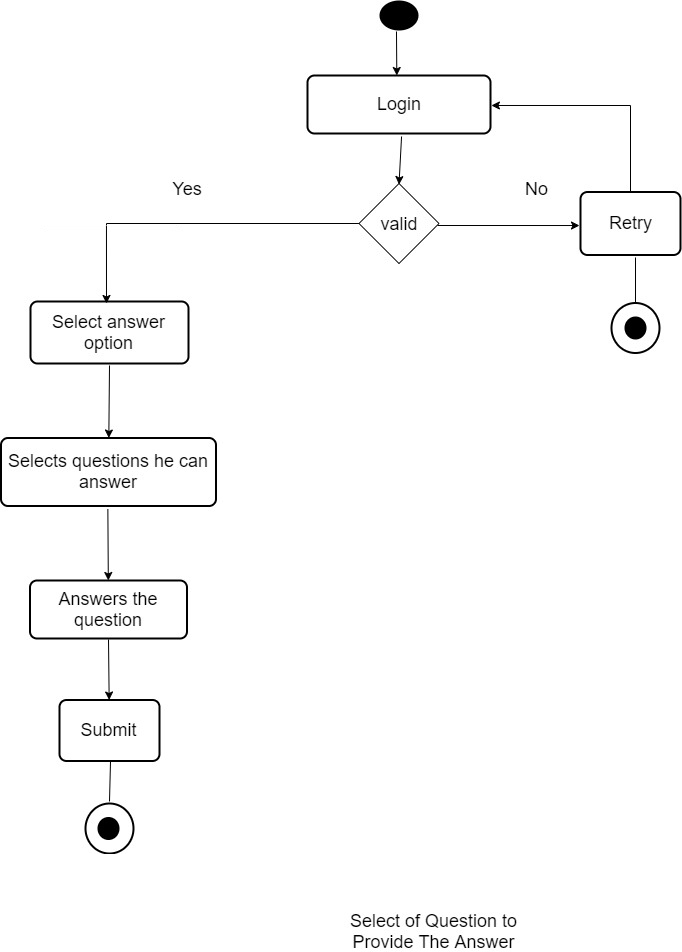


Fig 9: Activity Diagram 2: Answer a Question

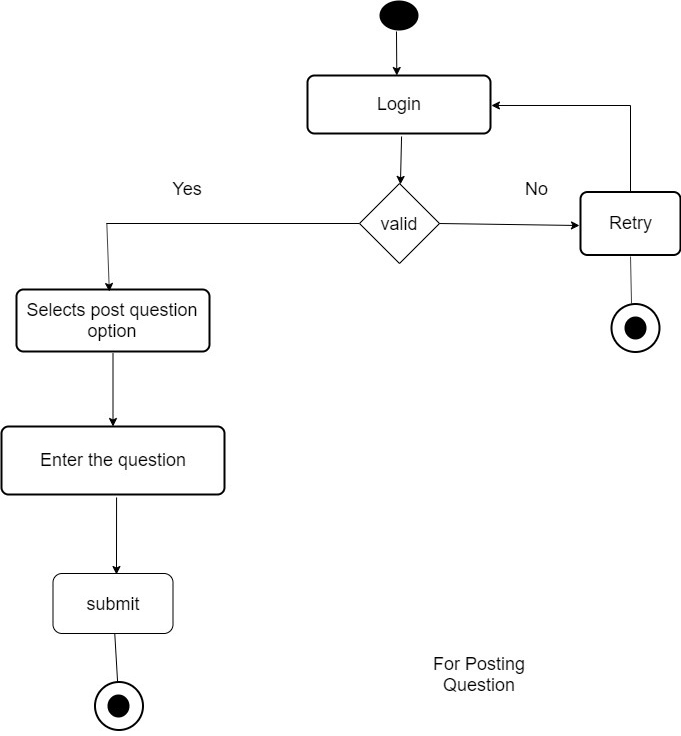


Fig 10: Activity Diagram 3: Ask a question

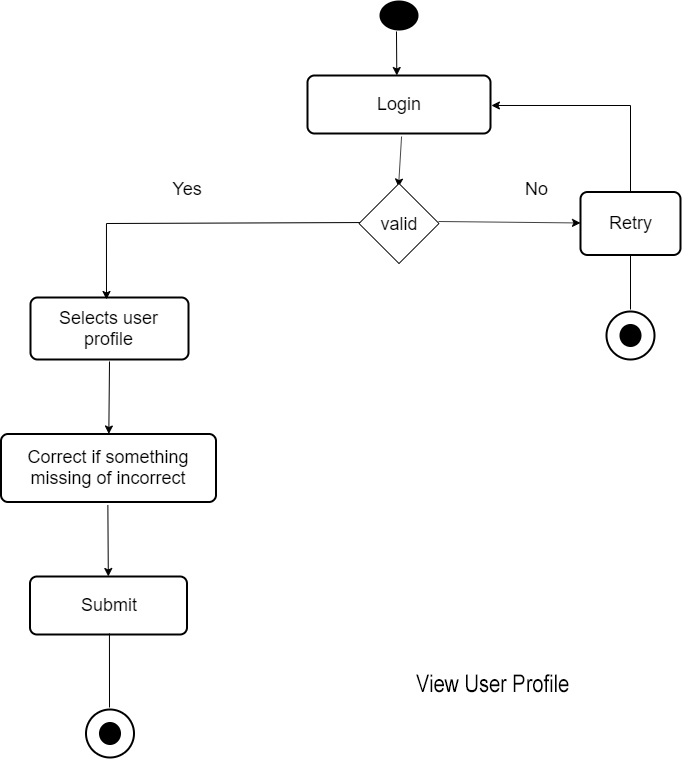
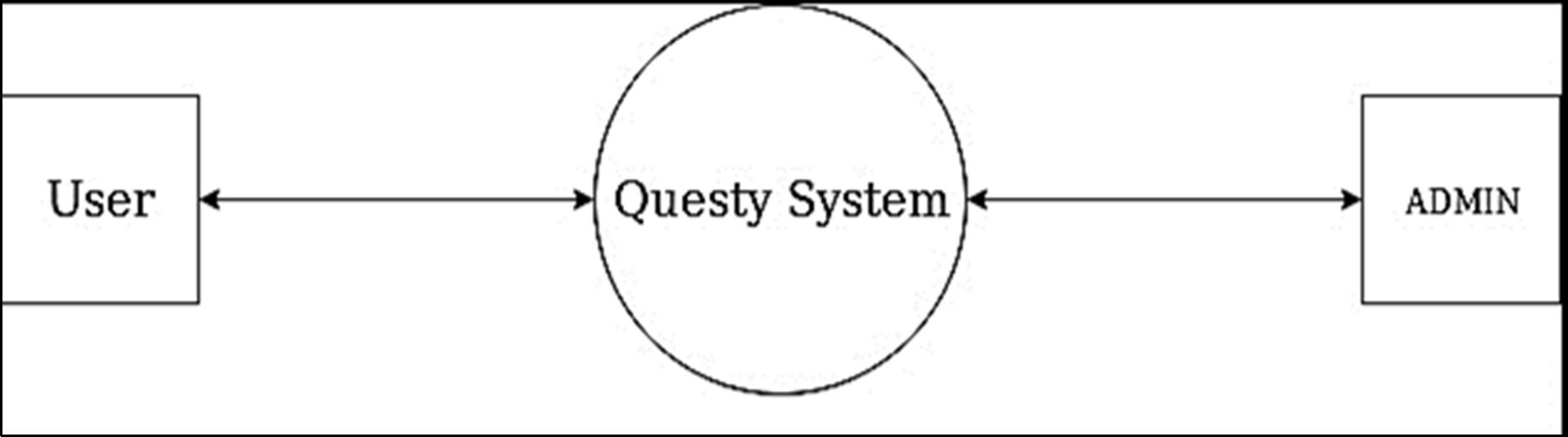
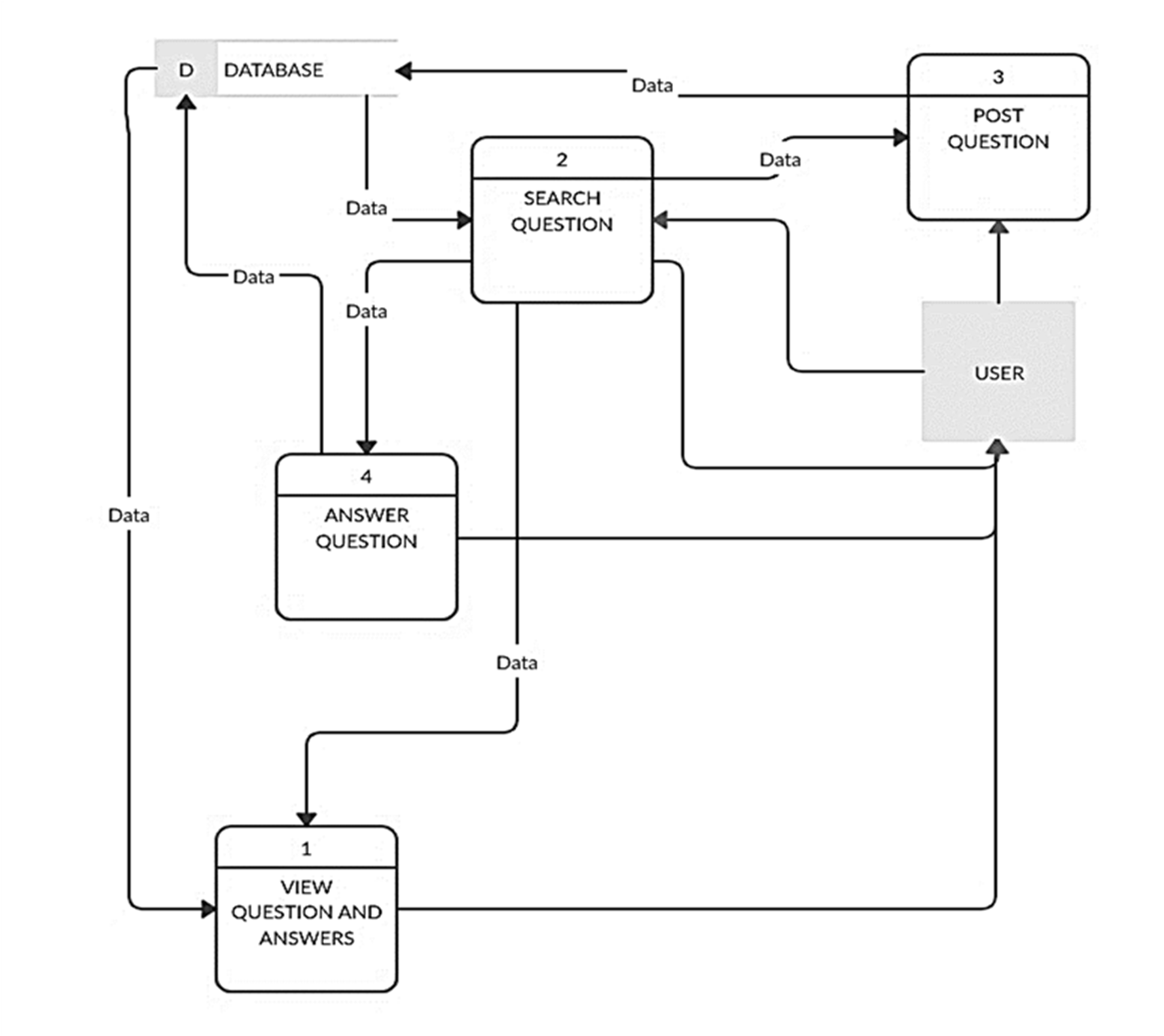


Fig 11: Activity Diagram 4: View User Profile

1. Data Flow Diagram: A data flow diagram shows the way information flows through a process or a system. It includes data input and outputs, data stores, and the various sub-processes that the data moves through. DFDs are built using standardized symbols and notations to describe various entities and their relationships.



1. Level 0



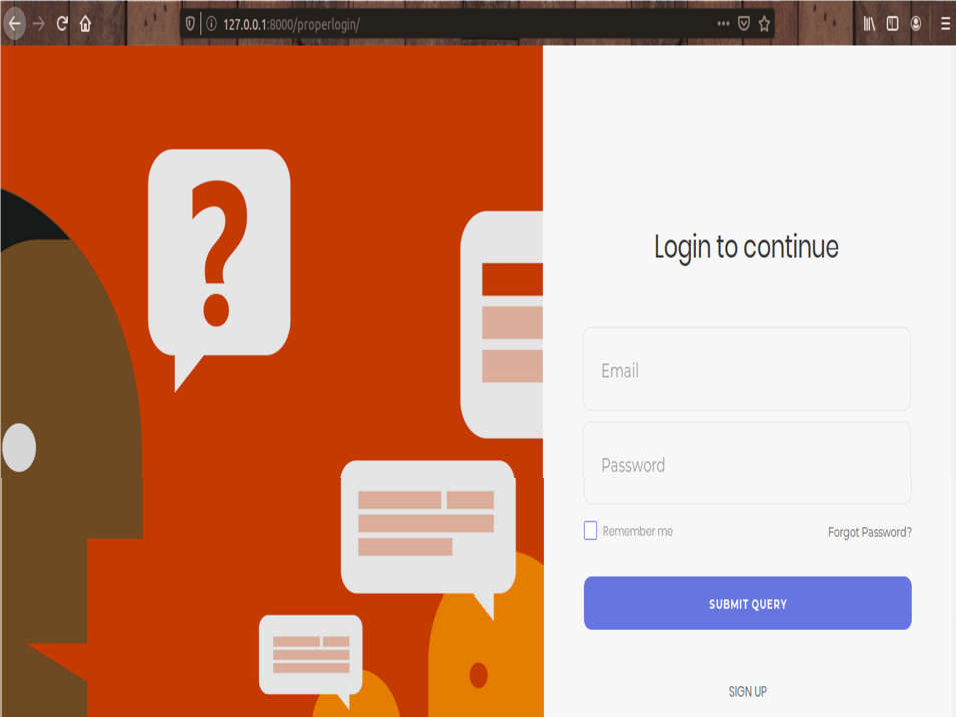
1. Level 1

Fig 12: Data Flow Diagram

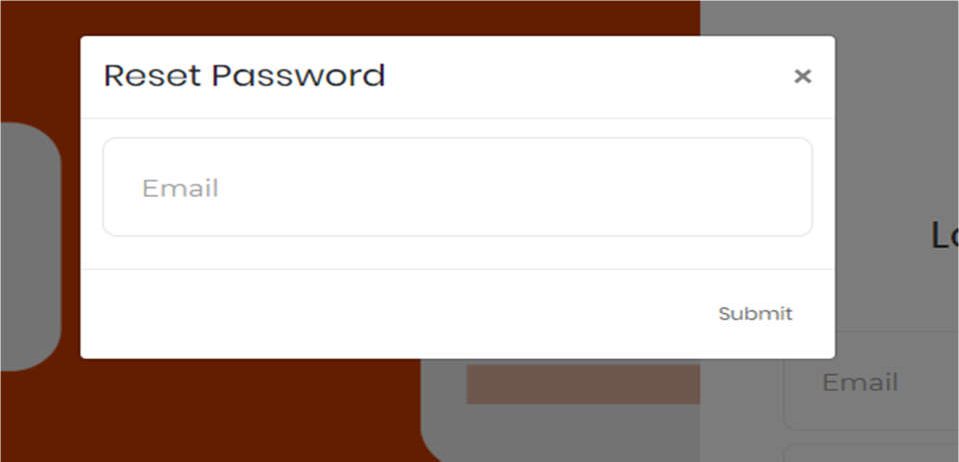
# IMPLEMENTATION AND TESTING

**Login page:**

In order that only authentic users can look for answers, post questions, answer queries and search for things we have implemented a compulsory login portal. A Login form is used to enter authentication credentials to access a restricted page or form. The login form contains a field for the username and another for the password. When the login form is submitted its underlying code checks that the credentials are authentic, giving the user can access the restricted page. If a user is not able to provide authentic credentials they will not be able to proceed past the login form.

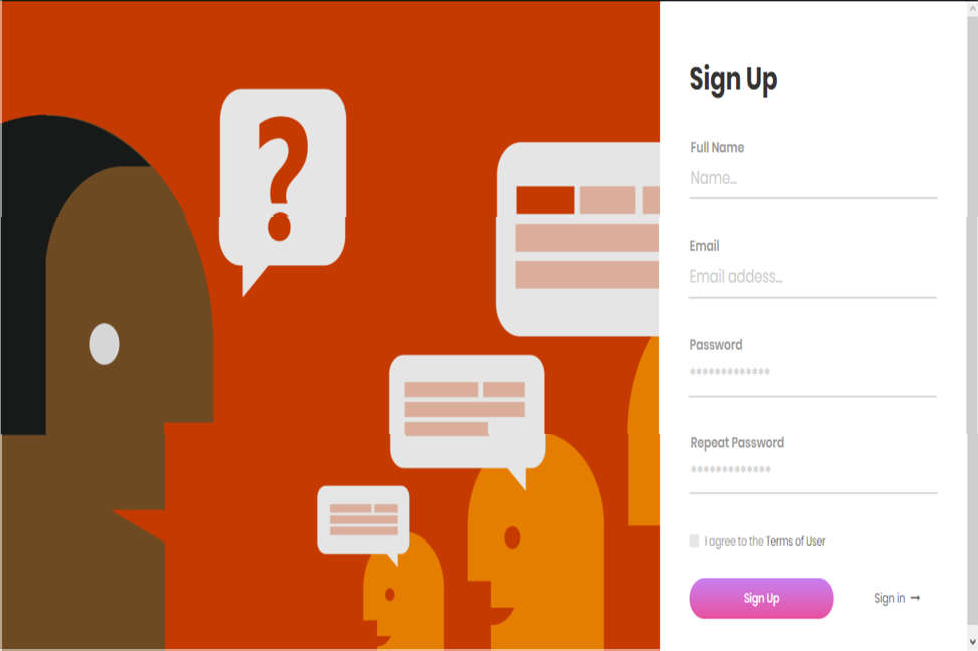


In this login form, we have made use of almost all the development tools described above. The login page also has an option for forgot password which is implemented using modals inbuilt in Bootstrap library.



**Sign Up:**

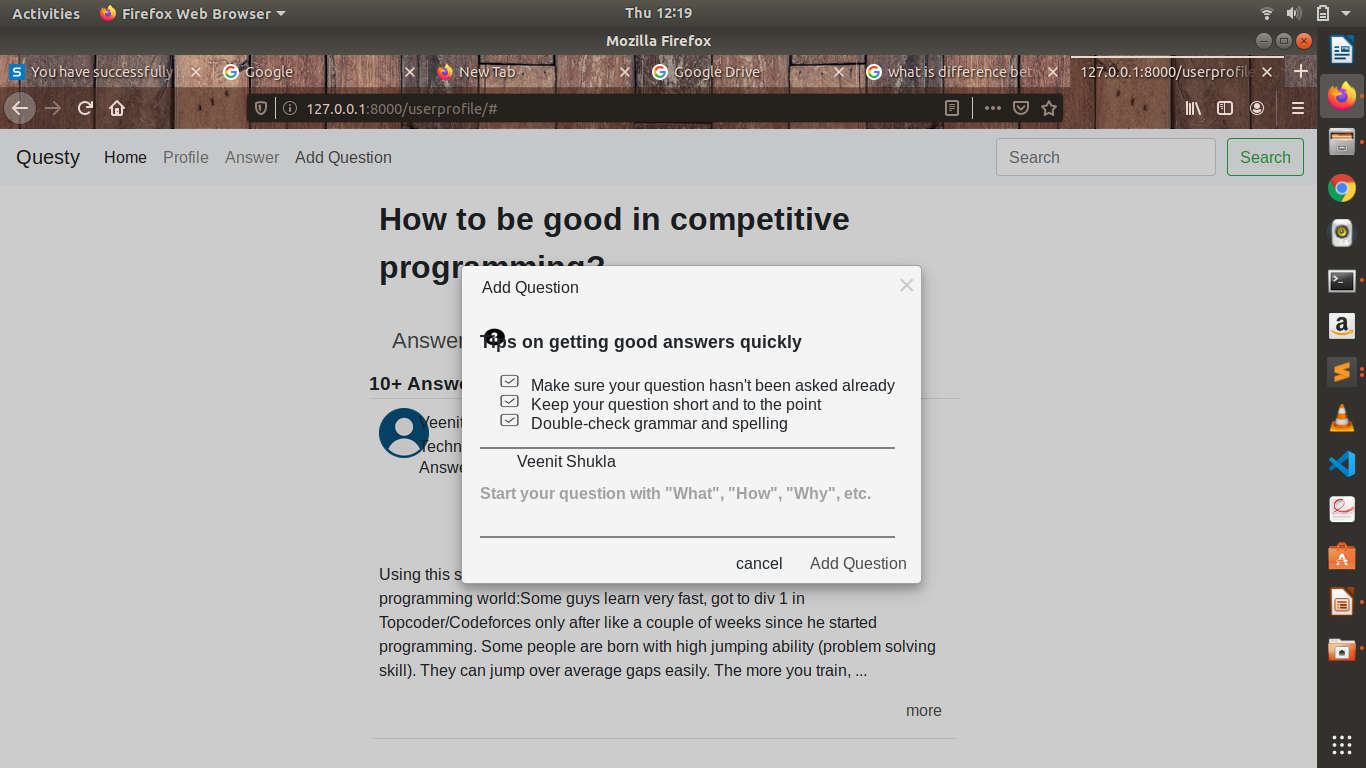
For new users, we also have the option to register on our forum to ask questions, post answers and also look for queries. The login page itself has the option to register for Questy.



**Home page:**

Once a user has logged in he will reach the home page of our project. The homepage has the view of the top questions and their answers in shorthand form. Also, they will be able to view the unanswered questions from the home page. The search bar is also available to search for any questions that the user wishes to search. Also under the profile tab, the user will be able to view, edit the questions and answers posted by himself.

**Add Question:**



Under this option, the user will have the option to add a question to our forum “Questy”. The modal used will also give us a brief view of the question outline so it is helpful to others as well.

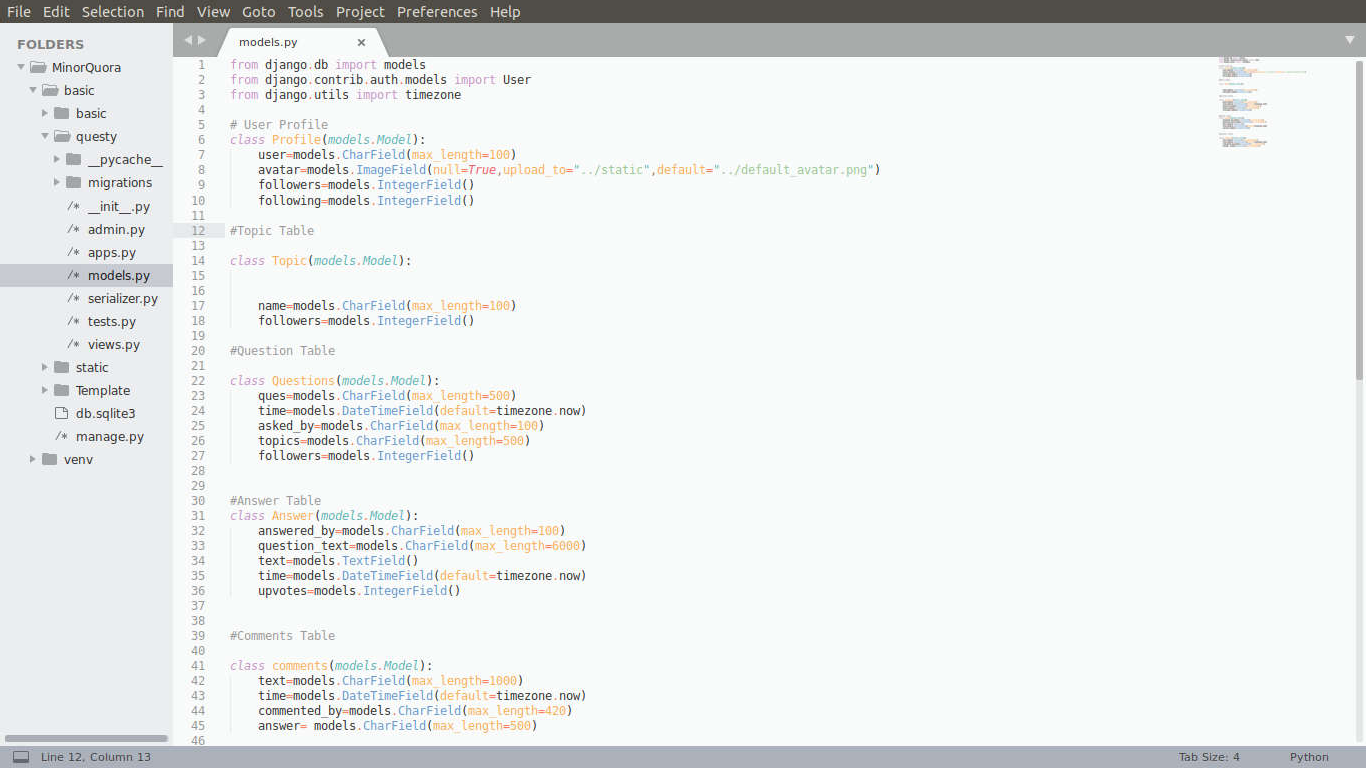
**Application of models**

A model is the single, definitive source of information about your data. It contains the essential fields and behaviors of the data you’re storing. Generally, each model maps to a single database table.

The basics:

* Each model is a Python class that subclasses django.db.models.Model.
* Each attribute of the model represents a database field.
* With all of this, Django gives you an automatically-generated database- access API; see Making queries.

Once you have defined your models, you need to tell Django you’re going to use those models. Do this by editing your settings file and changing the INSTALLED\_APPS setting to add the name of the module that contains your models.py. The most important part of a model – and the only required part of a model – in the list of database fields it defines. Fields are specified by class attributes. Be careful not to choose field names that conflict with the models API like clean, save, or delete.



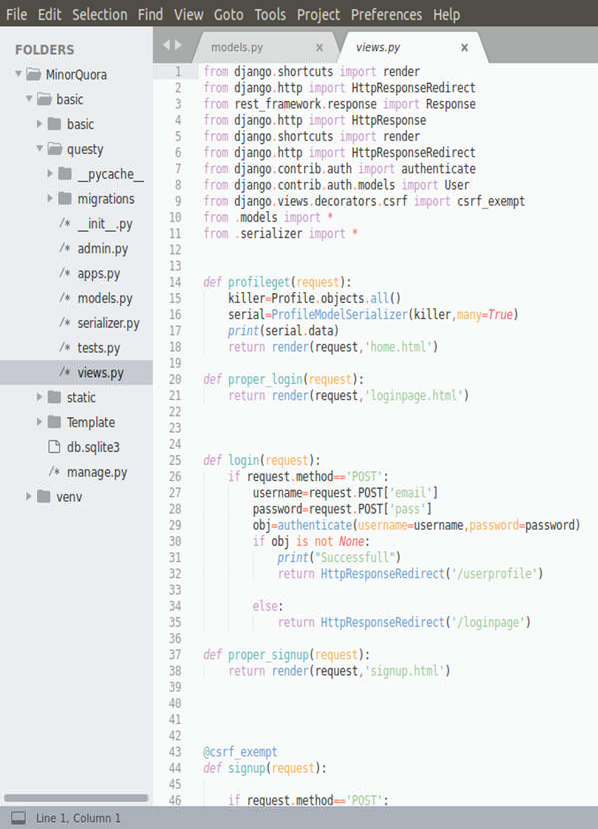
**Working on views.py**

A view function, or view for short, is simply a Python function that takes a Web request and returns a Web response. This response can be the HTML contents of a Web page, or a redirect, or a 404 error, or an XML document, or an image. The view itself contains whatever arbitrary logic is necessary to return that response.

* + First, we import the class HttpResponse from the Django.http

module, along with Python’s datetime library.

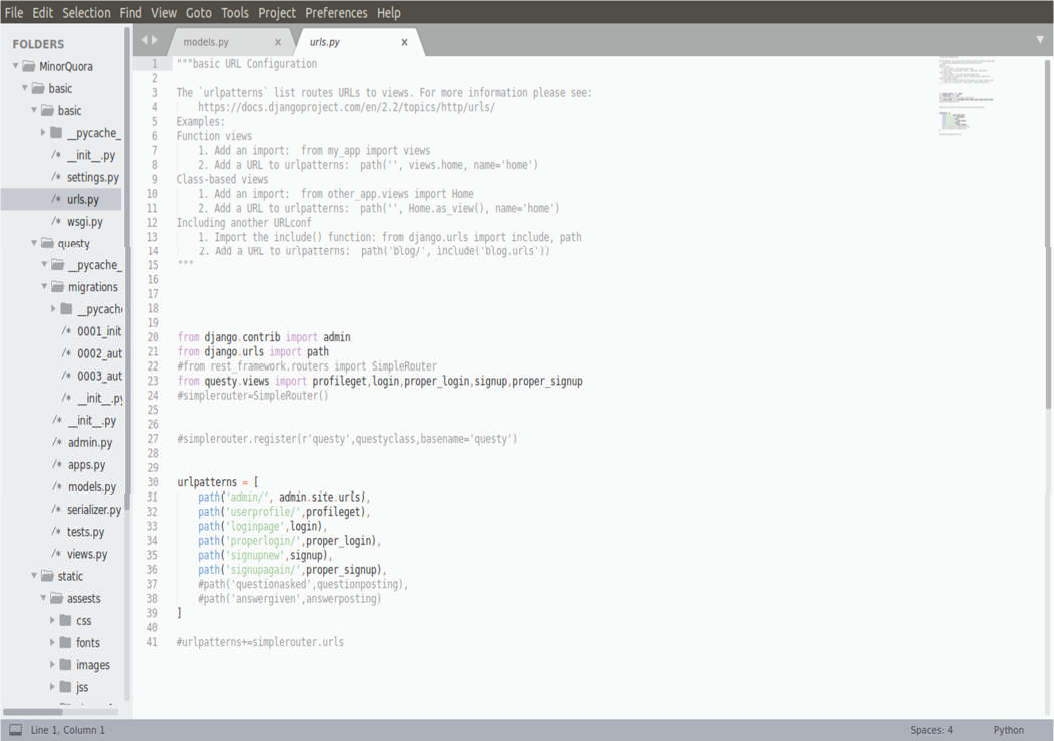
* + Next, we define a function called current\_datetime. This is the view function. Each view function takes an HttpRequest object as its first parameter, which is typically named request.
  + Note that the name of the view function doesn’t matter; it doesn’t have to be named in a certain way in order for Django to recognize it. We’re calling it current\_datetime here because that name clearly indicates what it does.
  + The view returns an HttpResponse object that contains the generated response. Each view function is responsible for returning an HttpResponse object. (There are exceptions, but we’ll get to those later.)



**Deployment using urls.py**

A clean, elegant URL scheme is an important detail in a high-quality Web application. Django lets you design URLs however you want, with no framework limitations.

To design URLs for an app, you create a Python module informally called a URLconf (URL configuration). This module is pure Python code and is a mapping between URL path expressions to Python functions (your views). This mapping can be as short or as long as needed. It can reference other mappings. And, because it’s pure Python code, it can be constructed dynamically. Django also provides a way to translate URLs according to the active language. See the internationalization documentation for more information.



# CONCLUSION

In today’s world with the boon of the Internet and the explosion of information, people today are more confused as to what to use this information for. Googling everything today isn’t practical as the results at times are more confusing than our problem itself. People, therefore, today are turning to Question and Answer forums for solutions to their problems. Be it technical, professional or even personal. People today seek to know how people respond to situations, doubts, and even struggles. At times it becomes really necessary to find out how people in the past who had been there had responded to some particular situation.

To provide a solution to this we built a website known as Questy. Where we seek to provide a platform similar to the giants like Quora, Stack Overflow, Yahoo answers, etc. We seek to help everyone with a question in his mind and also provide a platform for the experienced to share their thoughts.

We plan to further develop it into a discussion forum so that it can be used in universities, colleges and even among similar places for live discussions. We thank the University for allowing us to work on something like this. We have tried hard to provide as many features as we can in the short span of time that we had. We have tried to use the best possible tools available to make the interface and working as good as possible.

# REFERENCES

1. https://docs.python.org/3.7/
2. https[://ww](http://www.quora.com/)w.q[uora.com](http://www.quora.com/)
3. https://en.wikipedia.org/wiki/Web\_development
4. https[://ww](http://www.w3schools.com/)w.[w3s](http://www.w3schools.com/)c[hools.com/](http://www.w3schools.com/)
5. https[://ww](http://www.udemy.com/course/python-and-django-full-stack-web-)w.u[demy.com/course/python-and-django-full-stack-web-](http://www.udemy.com/course/python-and-django-full-stack-web-)

developer-bootcamp/learn/lecture/6598198?start=0#overview

1. https[://ww](http://www.youtube.com/channel/UCZc238wXqeN2M2uXQaY9MEw)w.y[outube.com/channel/UCZc238wXqeN2M2uXQaY9MEw](http://www.youtube.com/channel/UCZc238wXqeN2M2uXQaY9MEw)
2. https://docs.djangoproject.com/en/2.2/
3. https://getbootstrap.com
4. https://developer.mozilla.org/en-US/
5. https://stackoverflow.com

# STUDENT DETAILS

Name: Ishaak Mall Enrollment No: 171B054

Branch: Computer Science and Engineering Course: B.Tech.

Contact No: +91 8604809764 Email ID: [ishaak15@gmail.com](mailto:ishaak15@gmail.com)

Name: Veenit Shukla Enrollment No: 171B149

Branch: Computer Science and Engineering Course: B.Tech.

Contact No: +91 7355752272

Email ID: [veenitshukla20@gmail.com](mailto:veenitshukla20@gmail.com)

Name: Vichar Shukla Enrollment No: 171B150

Branch: Computer Science and Engineering Course: B.Tech.

Contact No: +91 9695592513

Email ID: [shuklavichar56@gmail.com](mailto:shuklavichar56@gmail.com)