

PROJECT REPORT

**PROJECT NAME : NEWS TRACKER APPLICATIONS – CLOUD
COMPUTING**

TEAM ID : PNT2022TMID30007

TEAM LEADER : ANKIT KUMAR JHA.A

TEAM MEMBERS : BALAJI.S

KEERTHIVASAN.R

LOKESH.G

ABSTRACT

Recently, the Internet usage spread in all areas of life. Online news is among the popular articles on the Internet, which occupies a large portion of online information. The online news will be viewed almost every second in order to follow the evolution of any desired global events. There are many organizations or political parties employ agents for tracking news by grouping the event. Therefore, news clustering is helpful and worthy for many researchers and online news readers in order to view events from multiple perspectives. Additionally, it can be used in online news summarization, topic detection and tracking for extracting and detecting new events or topics in the news articles. The news extraction can be applied on news articles in the form of monolingual or multilingual. On the other hand, news aggregation is the most important method for scrawling and collecting events based on topics or categorization. This paper investigates the challenges and issues that relate to online news research. The discussions include the overview of system architectures, online news techniques, and a few related computer applications for the above mentioned online news areas. The main purpose of this project is to identify and follow tasks which we need only specific information news sources. We are going to use python code based on advanced cloud technology application for detecting the topics and to calculate the similarity of topics.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Digital news continues to evolve, encouraged by a various innovation in recent time, from groundbreaking new technologies like virtual reality and automated reporting to experiments on social platforms that have altered campaign coverage. Topic detection and tracking is a challenging topic in information retrieval technology that can be used in the text mining. In topic detection we finding the most important topics in a collection of news articles of which only people prefer specific news. Our approach combines a variety of learning techniques. Topic detection is an unsupervised task and topic tracking is supervised task of need only topics to save their time.

We are going to use python coding to create topic clusters and pyqt5 classifier for tracking topics. To identify the serious news, we identify the clusters that fall into same category. Newspapers normally receive the news from various news agencies with very few changes. Thus, the corpus of news articles contains the same events written by different journalists which must be eliminated from the collection and contains lots of information which may take time for everyone to check and read specific article for which they actually need.

The main purpose of research is to identify interesting events happening in the world and to give the specific topic for which people are interested. Analysts are continuously trying to identify latest news and story from a very large resources of information that arrives daily and need one article. Information of news is been collected at anywhere through the cloud technology.

1.2 OBJECTIVE

- The main objective of this project is to save time and not to waste their timing on reading all the articles from the newspaper.
- Text mining utilizes techniques from the field of data mining, combines methodologies from various other areas such as categorization, important information retrieval, clustering, summarization, information extraction, computational linguistics, concept linkage and topic tracking.
- Can track their information of news for only which they need.
- We can collect data from anywhere through the cloud computing method.

CHAPTER 2

LITERATURE SURVEY

[1] Aksoy, F. Can, and S. Kocberber: Novelty Detection for Topic Tracking, vol. 63, no. 4, pp. 777–795 (2021)

Allan extract new tasks and then track the topic similar as an information filtering task which was done by querying new documents against the profile of the newly detected topic.

[2] Pouliquen, R. Steinberger, C. Ignat, E. Käsper, and I. Temnikova: Multilingual and cross-lingual news topic tracking (2019).

Topics representation was done using vector of stemmed words and their TF-IDF values, considering only nouns, adjectives, verbs etc. In their results the produced 10 and 20 features optimal results. Observed that the identification and usage of proper geographical references, names, significantly improves document similarity calculation and clustering. Hyland put news in clusters and detected topics manipulating the different combinations of various named entities to link related documents.

[3] Schultz, J. Michael, and Mark Liberman: Topic detection and tracking using idf-weighted cosine coefficient. In Proceedings of the DARPA broadcast news workshop, pp. 189-192. San Francisco: Morgan Kaufmann (2019).

Topic detection is an unsupervised and topic tracking is supervised. In our approach, we are going to use hierarchical agglomerative clustering for topic detection based on average linkage using document similarity vector. For document similarity, we have to use cosine similarity based on TF-IDF

[4] Juha Makkonen : Semantic Classes in Topic Detection and Tracking (2019).

In our approach first step is preprocessing on collected text news. A corpus consists of a large data which is collected from different sources. Due to heterogeneity, data becomes noisy and inconsistent. Due to inconsistency in data, mining process can lead to confusion and hence result becomes inaccurate. So to extract consistence and accurate data pre-processing is applied to the data. Data Pre-processing enhances the quality of data and at the same time reduces the difficulty of the mining process.

[5] Bijal, Dalwadi, and Suthar Sanket: Overview of stemming algorithms for Indian and Non-Indian languages, arXiv preprint arXiv: 1404.2878 (2020).

The main purpose of stemming is to reduce different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root form. Stemming is widely uses in Information Retrieval system and reduces the size of index files.

[6] Makkonen, Juha: Semantic classes in topic detection and tracking (2019).

In this research work, we are dealing with texts. For finding similarity between texts we need to convert texts into vectors, so we are going to use VSM model. We are using the SVM model as the baseline system because it is easy to implement, robust and more competitive compared to other the additional elaborate systems.

CHAPTER 3

EXISTING SYSTEM AND PROPOSED SYSTEM

3.1 EXISTING SYSTEM

Topic detection is an unsupervised and topic tracking is supervised. Data Pre-processing enhances the quality of data and at the same time reduces the difficulty of the mining process. The word with higher score has more important in the document. Single NEWS which contains seven different keywords which describes its content, so the story will appear in seven different clusters.

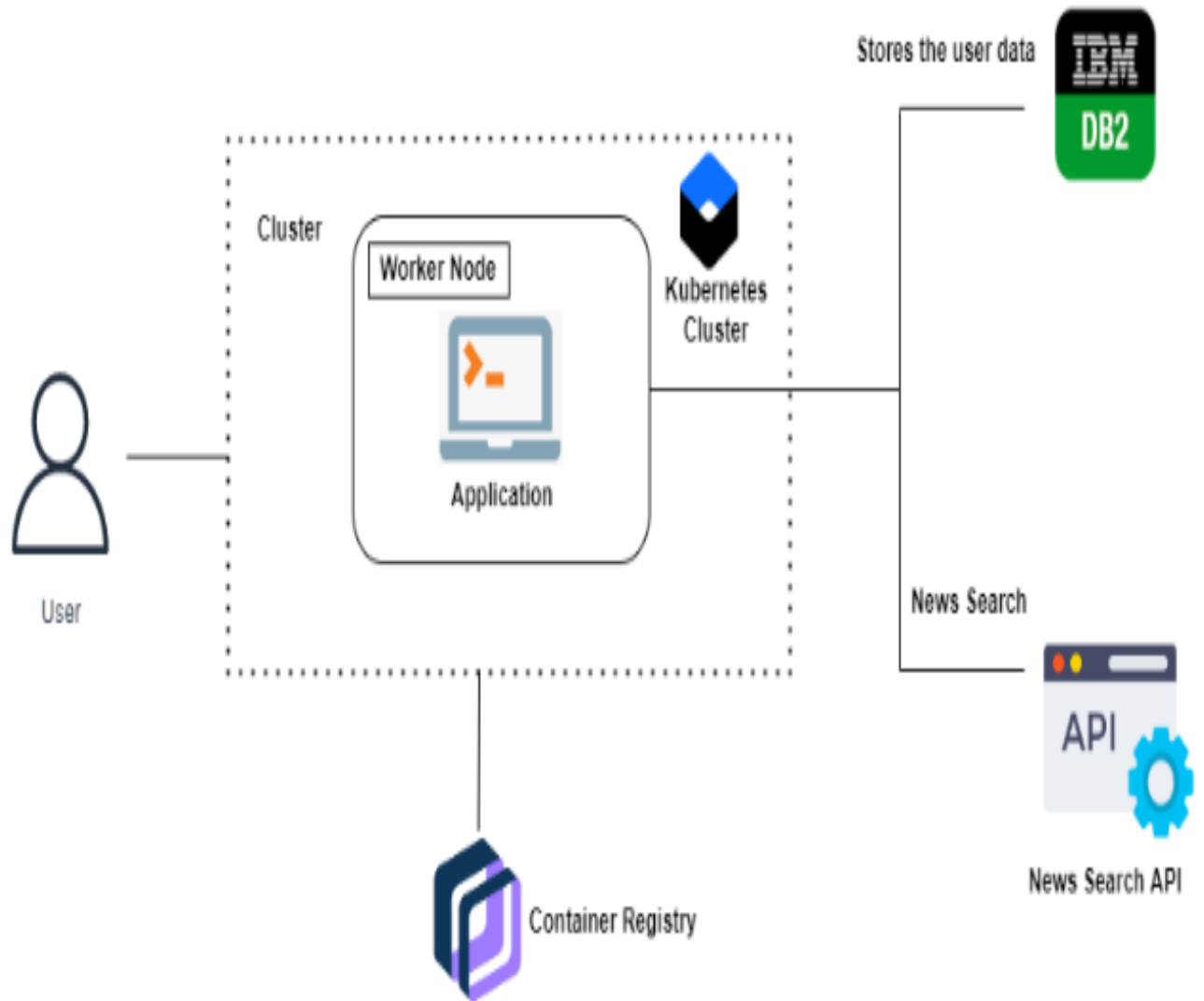
There are multiple news-sharing apps used by a single user and are often spammed with notifications. There is also a lot of fake news which gets shared. News organizations whose mobile apps only provide users with their articles or videos are missing a big opportunity. An application, by definition, should be applied to perform a task, to solve a problem. Most news doesn't do that.

3.2 PROPOSED SYSTEM

When we open the login page, if we are new to the app, then we have to register it in the app application, by entering some details of user name, email and password. The registered data will be stored in cloud data, that can be accessed at anytime and anywhere. Once after completing login page. They can search the article of news they want. Through the dashboard they can easily check the news for which they want and mostly news article which is given will be the updated one and referred one. Through this we can easily identify and come up with news information that we need. This can save our time a lot. This is user friendly and reliable to use. Here we use cloud technology which is the most advanced technology.

CHAPTER 4

4.1 ARCHITECTURE DESIGN



CHAPTER 5

SYSTEM SPECIFICATION

SOFTWARE SPECIFICATION

Front End : pyqt5
Back End : python
Operating System : Windows 10

HARDWARE SPECIFICATION

PROCESSOR : INTEL CORE I3.
RAM : 4 GB RAM
MONITOR : 15" COLOR
HARD DISK : 100 GB

5.3 MODULES

There are 5 types of modules in this project

1. User creation
2. User details
3. Displays of news
4. IBM cloud.

In the account creation module, the user can create „n“ number of accounts with individual maintenance.

5.4 MODULE DISCRPTION

A. User Creation

In the account creation module, the user can create „n“ number of accounts with individual maintenance.

B. User details

This module for user can enter their details such as name, address, mobile no, etc.,

C. Display of news

Through the rapid API we can access the news articles which we need

D. IBM cloud

In IBM cloud all the data has been stored through the web application developments. This cloud processes will store all the details of user and can be viewed in data viewed table.

5.5 Web applications

Web applications do not need to be downloaded since they are accessed through a network. Users can access a Web application through a web browser such as Google Chrome, Mozilla Firefox or Safari.

For a web app to operate, it needs a Web server, application server, and a database. Web servers manage the requests that come from a client, while the application server completes the requested task. A database can be used to store any needed information.

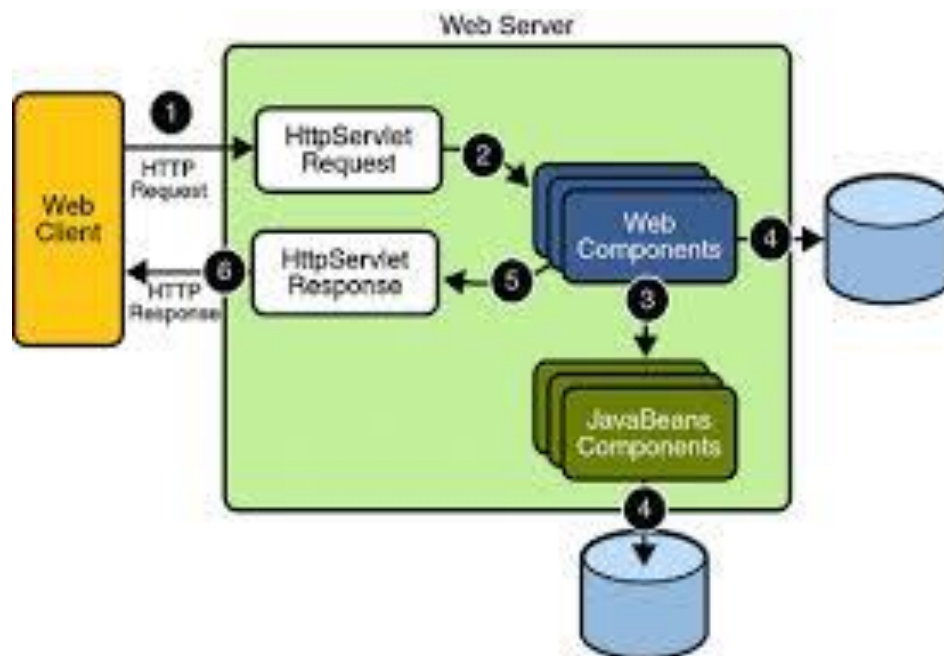
Web applications typically have short development cycles and can be made with small development teams. Most Web apps are written in python. Client-side programming typically utilizes these languages, which help build an application front-end. Server-side programming is done to create the scripts a Web app will use. Languages such as Python, Java, and Ruby are commonly used in server-side programming.

Web applications are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the

program executable. Some applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server.

The web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and, sometimes, a database to store the information.

- It executes at client side and all the resources of page loads at client side.
- It executes at the server side at the central server.
- It generates dynamic Web pages containing various types of markup language like HTML,XML
- It implements the end point of Web service.



Benefits

Web applications have many different uses, and with those uses, comes many potential benefits. Some common benefits of Web apps include:

- Allowing multiple users access to the same version of an application.
- Web apps don't need to be installed.
- Web apps can be accessed through various platforms such as a desktop, laptop, or mobile.
- Can be accessed through multiple browsers.
- Web applications run on multiple platforms regardless of OS or device as long as the browser is compatible
- All users access the same version, eliminating any compatibility issues
- They are not installed on the hard drive, thus eliminating space limitations
- They reduce software piracy in subscription-based web applications (i.e. SaaS)
- They reduce costs for both the business and end user as there is less support and maintenance required by the business and lower requirements for the end user's computer.

As a developer, mastering web-based applications opens many possibilities for you. Every company today requires an effective and affordable solution for software applications that allows them to reach out to users on every device, whether desktop or mobile, and on every platform. Web application development is platform-independent, which means that businesses don't have to create separate apps for iOS and Android, this can increase the price. Web application technology has developed tremendously in the past several years. They have made Web applications as immersive and personal as native applications. Even companies that have top-rated native apps require the same high-quality web applications since they can't afford to lose their web and desktop customers. Facebook, Instagram, WhatsApp all have web applications that are compatible that integrate with native applications. Learning

how to build an application for the web puts you at the top of the market that includes the top companies that require an extensive online presence.

Web Application vs. other application types

Within the mobile computing sector, Web apps are sometimes contrasted with native apps, which are applications that are developed specifically for a particular platform or device and installed on that device. However, the two are not mutually exclusive. Native applications are applications typically downloaded and made specifically for the type of device it is downloaded on. Native apps can commonly make use of the device-specific hardware, such as a GPS or camera on a mobile native app.

Programs that combine the two approaches are sometimes referred to as hybrid applications. Hybrid apps work similar to a Web app but are installed to the device as a native app would be. Hybrid apps can also take advantage of device-specific resources by using internal APIs. Downloaded native apps can sometimes operate offline; however, hybrid apps don't have this functionality. A hybrid app will typically share similar navigation elements as a Web app since they are based on Web apps.

CATEGORIES OF WEB APPLICATIONS

The categories of web application based on use are

- Social networking websites. EX:- Twitter.
- Online calendars. EX:-Google calendar
- Chat sites. EX:- Meebo, Omgpop
- Online shopping websites. EX:- Ebay, Myntra
- Online telephone directories. EX:- Just Dial
- Online dictionaries and web mail services

- Internet search engines.

ADVANTAGES OF WEB APPLICATIONS

1. Requires no disk space
2. Reduce business costs
3. Centralized data is secure and easy to backup
4. Integrate easily with other server-side grid procedure
5. Provides a cross platform compatibility

DISADVANTAGES OF WEB APPLICATIONS

1. Slower and runs over internet
2. Requires compatible web browsers
3. Lack of flexibility
4. Takes longer to develop as more complex
5. Depends on server delivering the application.

The types used in Web Applications

Generally, web applications are divided into five categories, each with distinct characteristics and distinctive features. Let's examine the five types of web apps.

1. Static Web Applications

So-called because of their structure and inflexibility static web applications are more of a collection of static websites that show the same information to each user. They are created with PyQt5. Because the pages of static web application development are created on the server's end there is no space in the application for customization.

Pages for static web apps can be a challenge to modify or update. You'll need to perform an entire page reload for any changes to be visible.

With all the above are static web application development ever being carried out currently?

Certain companies that need a basic web page that conveys vital business information that requires no input can benefit from a static application. For instance, personal portfolios and business contact pages, and company information pages could utilize this type of web application development to achieve immediate and cost-effective results.

2. Applications for dynamic web

Dynamic web applications like the name imply they are interactive and require both client and server-side programming. For instance, if you enter details into a web application and get different results, then that's an active website.

If you can make a search, set up an account for yourself, leave comments or engage in other dynamic ways, then you're using a web application.

Pages in dynamic web applications don't follow a set sequence of displays but are formed by input on the server or client-side. They use an application called a content management system, or CMS, which alters the contents of the app.

A variety of programming languages, like PHP and ASP, create an online application. The apps need websites, databases, and servers.

Although technically, all applications that aren't static websites are dynamic, a few more broad categories comprise the remaining portion of this list of applications.

3. Ecommerce apps

If you plan to sell or purchase things online, you'll need at a minimum an online application that can take orders and accept payment. This kind of app is known as an e-commerce app.

This raises the technical difficulty of creating an online application to a new stage. From keeping a database up-to-date of merchandise to managing offers, orders, and transactions, web applications involve more programming.

The creation of an experience that is similar to native app experiences is also crucial for web applications for e-commerce so that customers can have a seamless experience, regardless of the device they're on.

4. Portal Web Apps

A portal web application is an application on the web that permits users to log in to secure areas from the home page. Consider credit card payments for example. It could be an online shopping site or bill payment site, and when it's time to pick an account with a credit card to pay for a transaction, then you're asked to sign in to the web portal of the credit card company's application to complete the transaction. This is an example of web-based portals.

5. Content Management System or CMS Web App

CMS, also known as a content management system, CMS can allow even non-technical users to create and edit content online. From simple blogs to a larger website and more, CMS allows users to keep these resources up-to-date with the most current information.

Making an app that users who are not technical require specific technologies that should be reserved for professional web application developers.

6. Progressive Web Apps

Progressive web applications (PWAs) aren't simply the term used to describe web-based applications. There, in actuality the most significant leap forward in web-based applications development.

A PerfectionGeeks utilizes the most up-to-date and most efficient web frameworks and technology for application development to provide you with an app that blends the best of native mobile and web applications.

These applications behave as an accessible website that you can save to your home screen with no download and then access at any time, you'd like in just a single click.

Furthermore, a progressive app makes use of a wide range of web technologies, plugins, and APIs, giving you an experience comparable to native apps that are top of the line however it is much easier and faster to build than native applications.

A study has shown that the average app loses 20 percent of its users each step from identifying the app until they begin using it. The reason for this is that users have to find a native app from the store and download it. Once downloaded, they can get connected and then start using it.

CHAPTER 6

6.1 PYCHARM INTRODUCTION

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below

Code Completion

PyCharm enables smoother code completion whether it is for built in or for an external package.

SQLAlchemy as Debugger

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

Git Visualization in Editor

When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

Code Coverage in Editor

You can run **.py** files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

Package Management

All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

Local History

Local History is always keeping track of the changes in a way that complements like Git. Local history in PyCharm gives complete details of what is needed to rollback and what is to be added.

Refactoring

Refactoring is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.

User Interface of PyCharm Editor

The user interface of PyCharm editor is shown in the screenshot given below. Observe that the editor includes various features to create a new project or import from an existing project.



Fig no: PyCharm software

6.1.1 PYCHAM INSTALLATION

From the screenshot shown above, you can see the newly created project Demo and the **site-packages** folder for package management along with various other folders.

In this chapter, you will learn in detail about the installation process of PyCharm on your local computer.

Steps Involved

You will have to follow the steps given below to install PyCharm on your system. These steps show the installation procedure starting from downloading the PyCharm package from its official website to creating a new project.

Step 1

Download the required package or executable from the official website of PyCharm <https://www.jetbrains.com/pycharm/download/#section=windows> Here you will observe two versions of package for Windows as shown in the screenshot given below –



Fig no: PyCharm installation

Note that the professional package involves all the advanced features and comes with free trial for few days and the user has to buy a licensed key for activation beyond the trial period. Community package is for free and can be downloaded and installed as and when required. It includes all the basic features needed for installation. Note that we will continue with community package throughout this tutorial.

Step 2

Download the community package (executable file) onto your system and mention a destination folder as shown below



Fig no: PyCharm community package

Step 3

Now, begin the installation procedure similar to any other software package.

Step 4

Once the installation is successful, PyCharm asks you to import settings of the existing package if any.



Fig no: PyCharm package

This helps in creating a new project of Python where you can work from the scratch. Note that unlike other IDEs, PyCharm only focusses on working with projects of Python scripting language.

This chapter will discuss the basics of PyCharm and make you feel comfortable to begin working in PyCharm editor.

When you launch PyCharm for the first time, you can see a welcome screen with entry points to IDE such as

- Creating or opening the project
- Checking out the project from version control
- Viewing the documentation
- Configuring the IDE



Fig no: PyCharm configure page

Recall that in the last chapter, we created a project named **demo1** and we will be referring to the same project throughout this tutorial. Now we will start creating new files in the same project to understand the basics of PyCharm Editor.

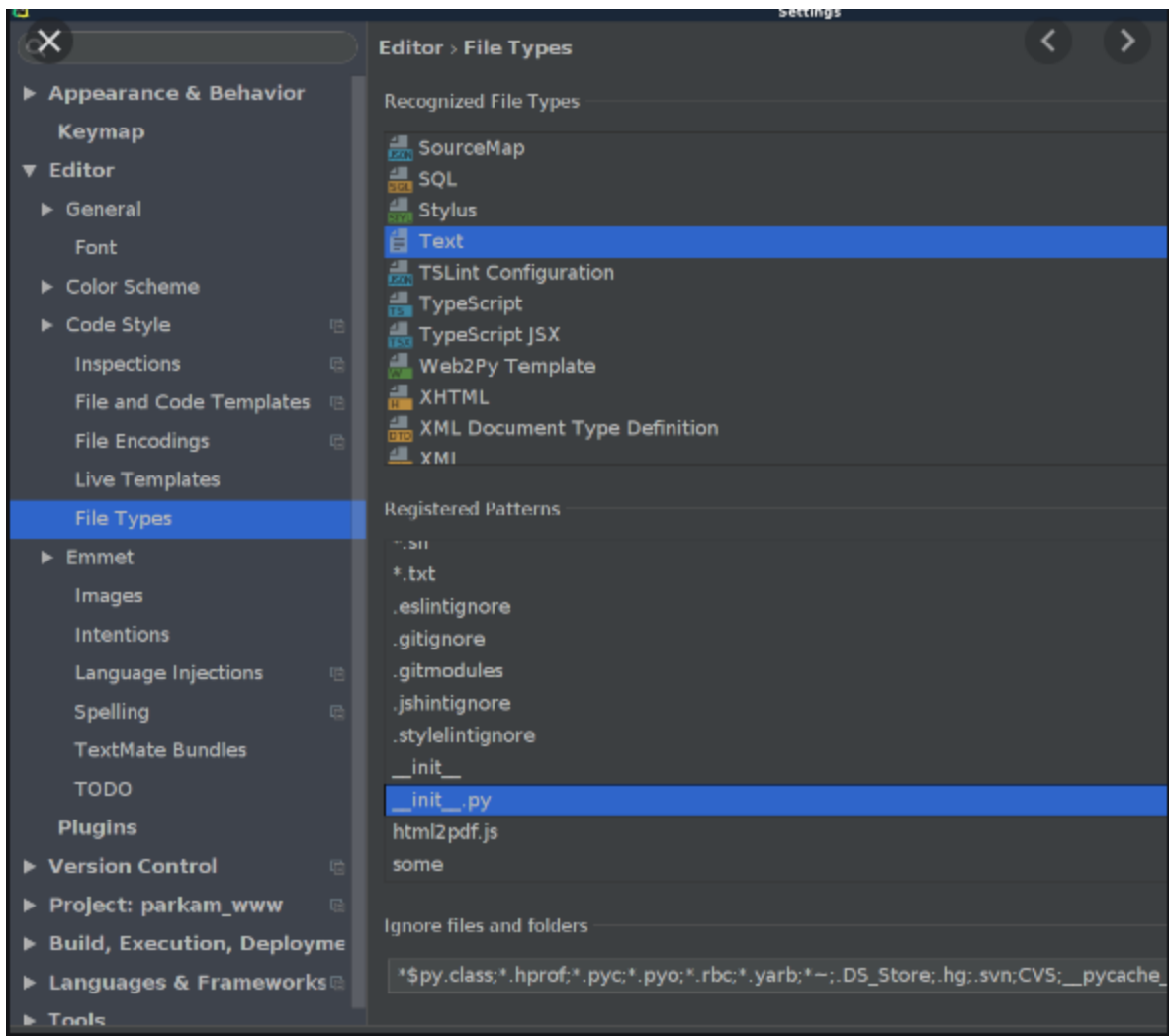


Fig no: Editing page

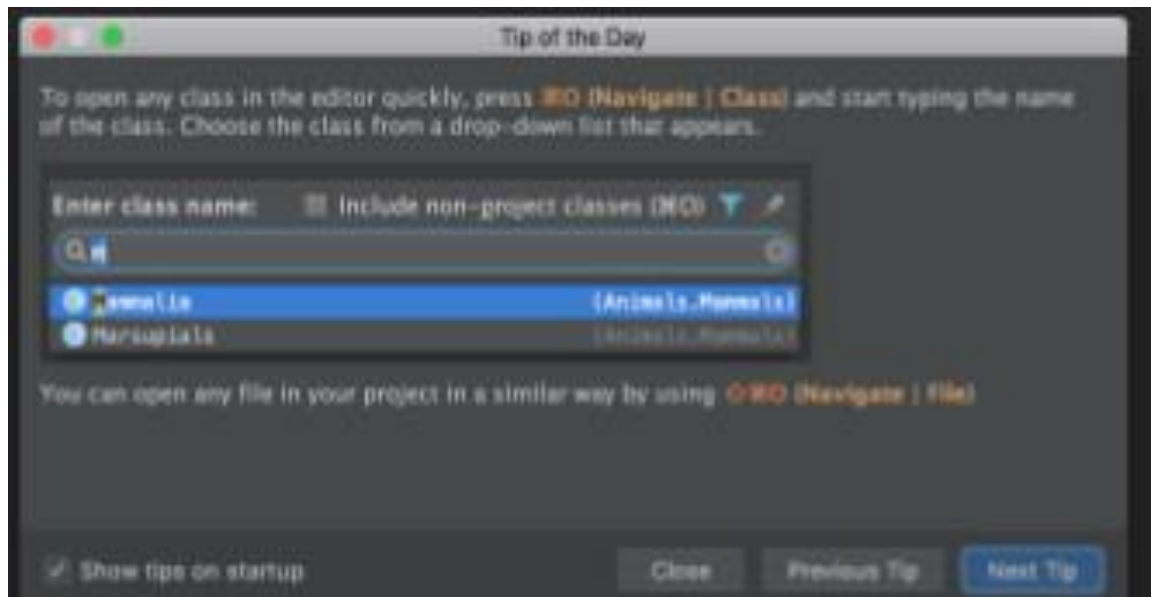


Fig no: Editor page

The above snapshot describes the project overview of demo1 and the options to create a new file. Let us create a new file called **main.py**.

The code included in main.py is as follows

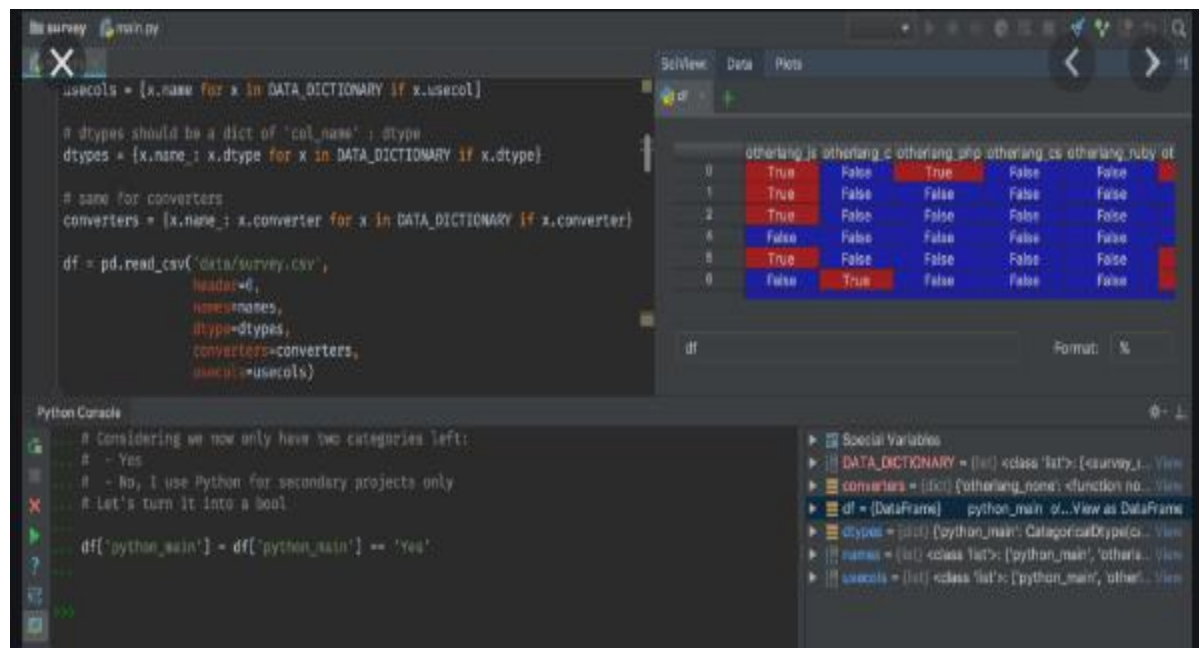
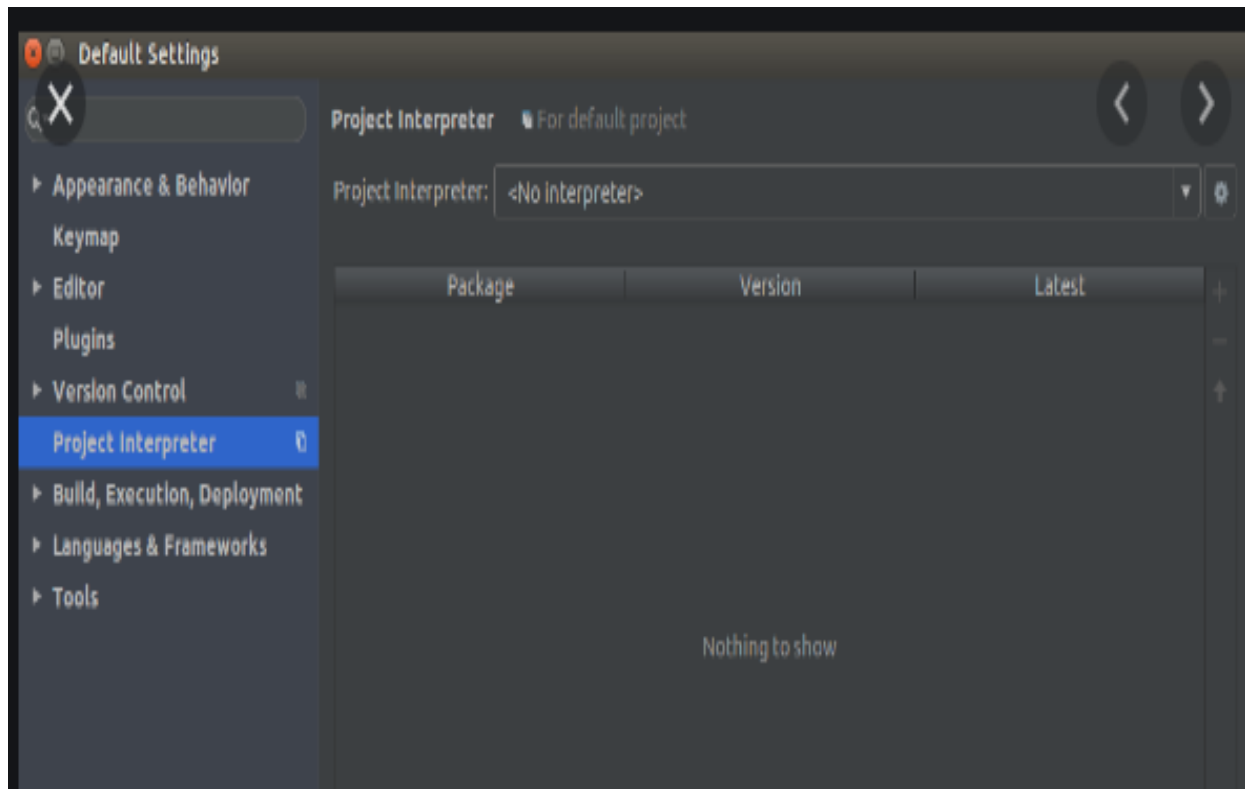


Fig no: coding window

The code created in the file **main.py** using PyCharm Editor is displayed as shown below

This code can be run within IDE environment. The basic demonstration of running a program is discussed below



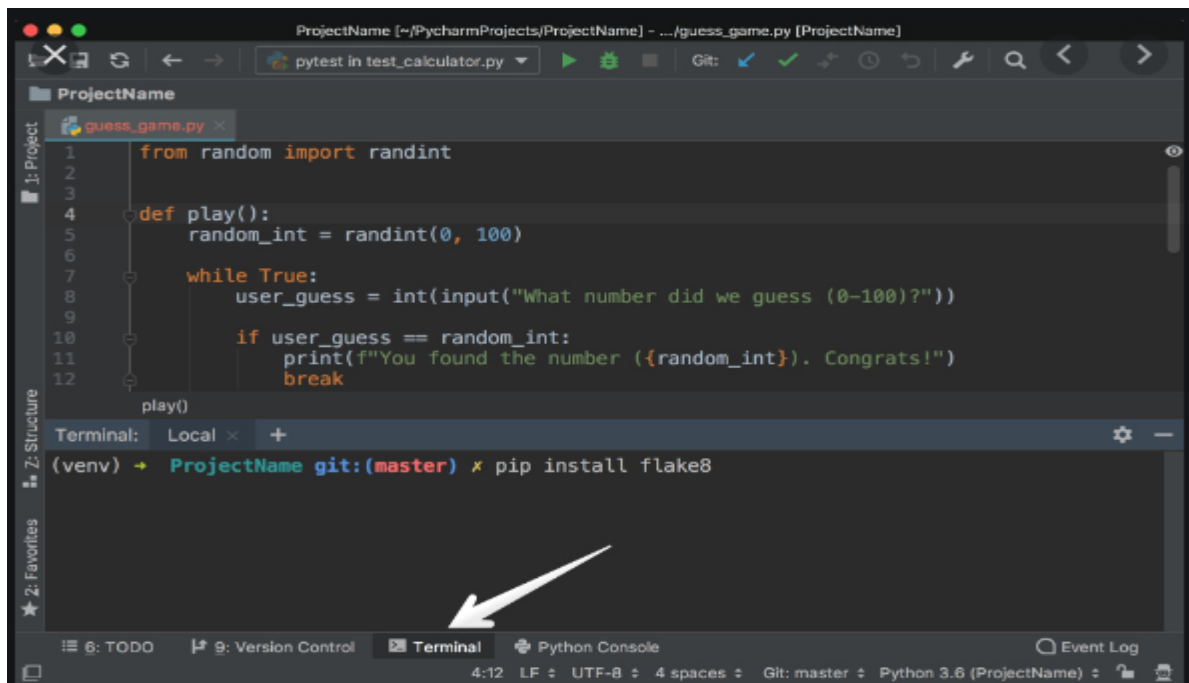


Fig no: coding window page

Note that we have included some errors within the specified code such that console can execute the code and display output as the way it is intended to.

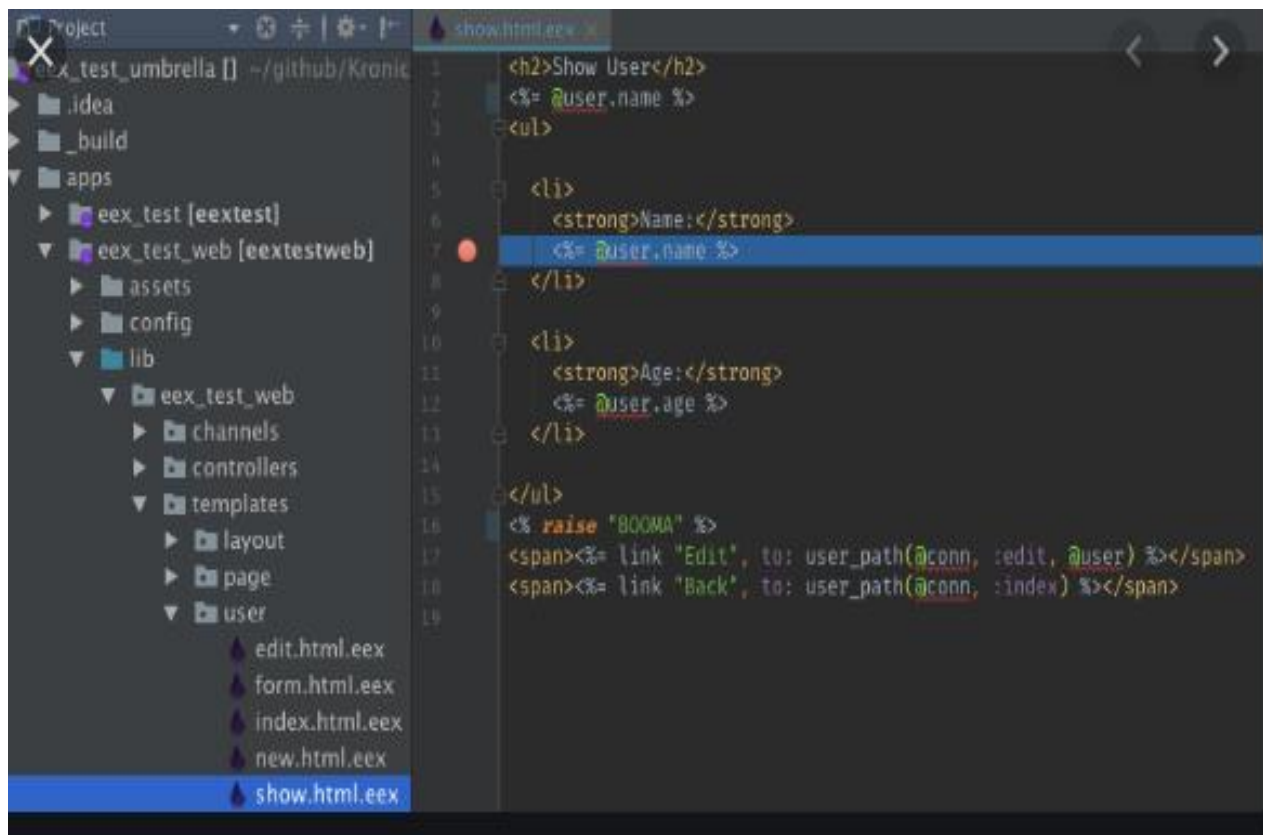


Fig no: Programming page

6.1.2 RUNING AND DEBUGGING

Running a python code comprises of two modes: running a script and debugging the script. This chapter focusses on debugging the Python script using PyCharm.

Steps Involved

The steps for debugging the Python project are as explained below –

Step 1

Start with debugging the Python project as shown in the screenshot below –

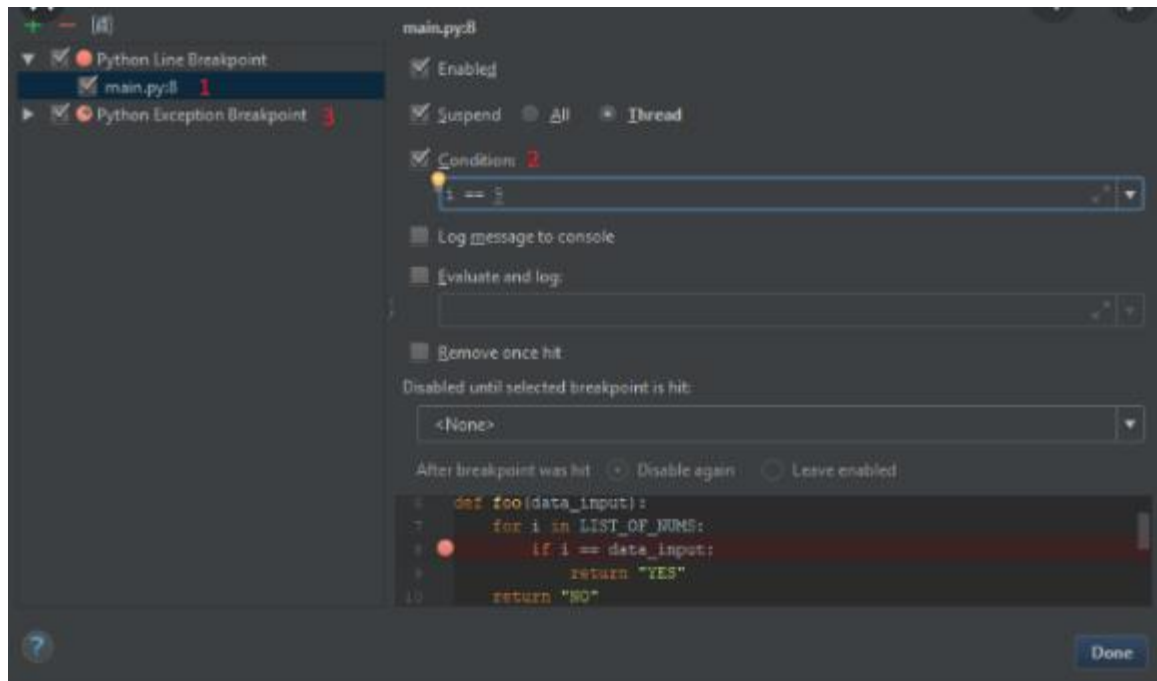


Fig no: Debugging window

Step 2

Now, Windows firewall asks permission for debugging the Python project as the procedure involves line by line compilation.

Step 3

The debugging console is created in PyCharm editor as shown below which executes the output line by line.

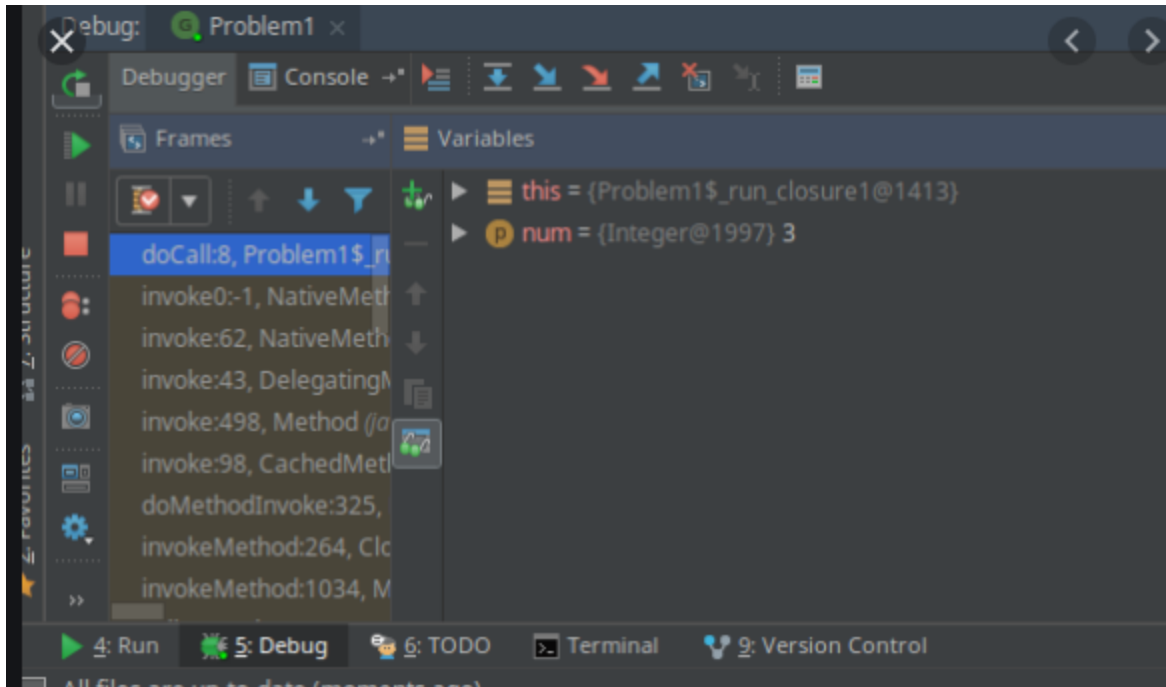


Fig no: Debugging console window

The run button moves from one line to another to execute the output as the way we want.

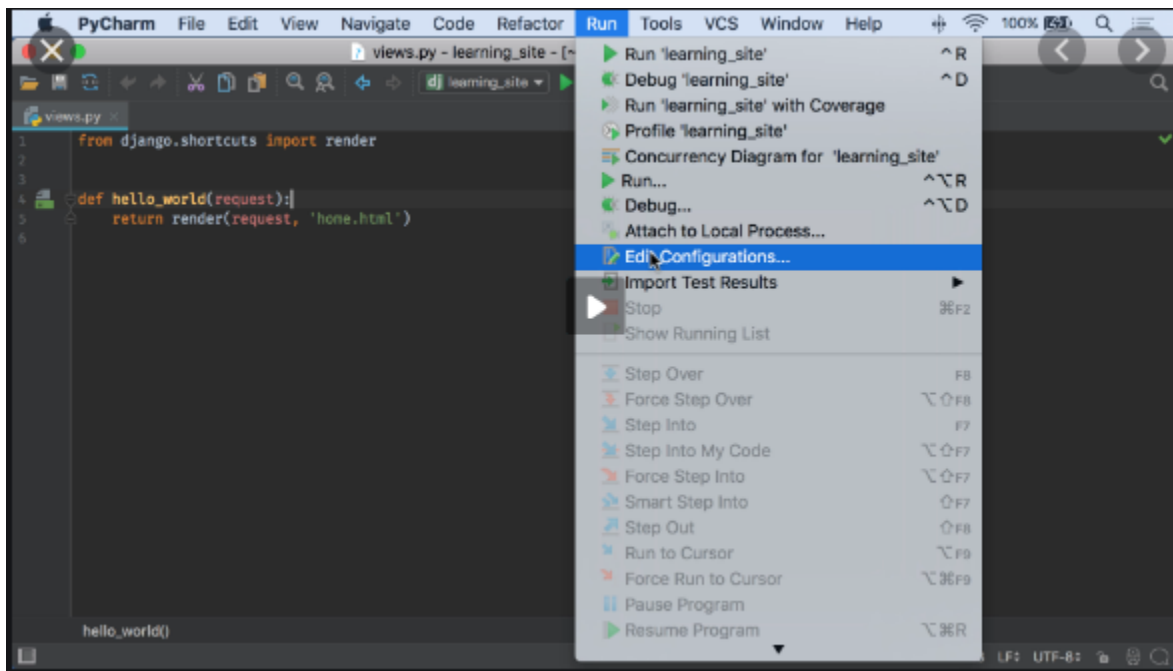


Fig no: Running window

Understanding Breakpoints

While debugging a particular script, it is intentional to create a breakpoint. Breakpoints are intentional stopping place or the place where the code is paused in order to identify the output at specific stage.

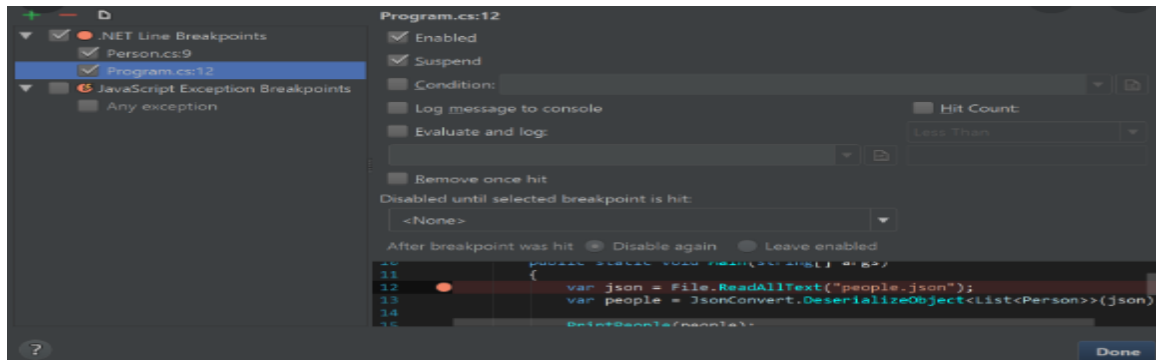


Fig no: Understanding Breakpoints

In PyCharm, breakpoints are visible using a separate dialog in the specified editor. It includes various attributes to evaluate the breakpoints defined and tracing log for the same with a main motive to achieve better programming practice.

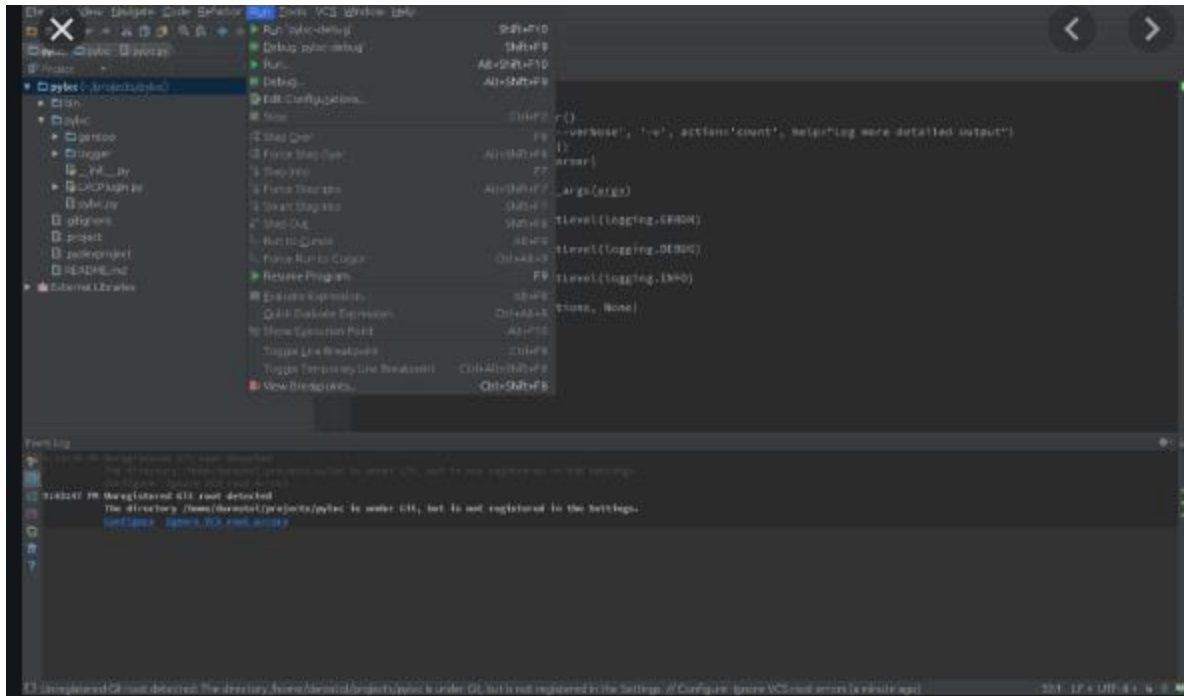


Fig no: Understanding Breakpoints window page

6.5 PYTHON

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions: **Python 2** and **Python 3**. Both are quite different.

Beginning with Python programming:

1) Finding an Interpreter:

Windows: There are many interpreters available freely to run Python scripts like IDLE (Integrated Development Environment) that comes bundled with the Python software downloaded from

Linux: Python comes preinstalled with popular Linux distros such as Ubuntu and Fedora. To check which version of Python you're running, type "python" in the terminal emulator. The interpreter should start and print the version number.

macOS: Generally, Python 2.7 comes bundled with macOS. You'll have to manually install Python 3 from

Let's analyze the script line by line.

Line 1: [# Script Begins] In Python, comments begin with a #. This statement is ignored by the interpreter and serves as documentation for our code.

Line 2: [print("GeeksQuiz")] To print something on the console, print() function is used. This function also adds a newline after our message is printed(unlike in C). Note that in Python 2, "print" is not a function but a keyword and therefore can be used without parentheses. However, in Python 3, it is a function and must be invoked with parentheses.

Line 3: [# Script Ends] This is just another comment like in Line 1.

Python designed by Guido van Rossum at CWI has become a widely used general-purpose, high-level programming language.

Prerequisites:

Knowledge of any programming language can be a plus.

Reason for increasing popularity

1. Emphasis on **code readability**, **shorter codes**, ease of writing

2. Programmers can express logical concepts in **fewer lines** of code in comparison to languages such as C++ or Java.
3. Python supports **multiple** programming paradigms, like object-oriented, imperative and functional programming or procedural.
4. There exists inbuilt functions for almost all of the frequently used concepts.
5. Philosophy is “Simplicity is the best”.

LANGUAGE FEATURES

Interpreted

- There are no separate compilation and execution steps like C and C++.
- Directly *run* the program from the source code.
- Internally, Python converts the source code into an intermediate form called bytecodes which is then translated into native language of specific computer to run it.
- No need to worry about linking and loading with libraries, etc.

Platform Independent

- Python programs can be developed and executed on multiple operating system platforms.
- Python can be used on Linux, Windows, Macintosh, Solaris and many more.

Free and Open Source; Redistributable

High-level Language

- In Python, no need to take care about low-level details such as managing the memory used by the program.

Simple

- Closer to English language; Easy to Learn
- More emphasis on the solution to the problem rather than the syntax

Embeddable

- Python can be used within C/C++ program to give scripting capabilities for the program's users.

Robust:

- Exceptional handling features
- Memory management techniques in built

Rich Library Support

- The Python Standard Library is very vast.
- Known as the “**batteries included**” philosophy of Python ;It can help do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, email, XML, HTML, WAV files, cryptography, GUI and many more.
- Besides the standard library, there are various other high-quality libraries such as the Python Imaging Library which is an amazingly simple image manipulation library.

6.3 PYQT5

PyQt5 is the latest version of a GUI widgets toolkit developed by Riverbank Computing. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. PyQt5 is a blend of Python programming language and the Qt library. This introductory tutorial will assist you in creating graphical applications with the help of PyQt. Our tutorial on earlier version – PyQt4 is available here.

PyQt is a GUI widgets toolkit. It is a Python interface for **Qt**, one of the most powerful, and popular cross-platform GUI library. PyQt was developed by Riverbank Computing Ltd. The latest version of PyQt can be downloaded from its official website – riverbankcomputing.com

PyQt API is a set of modules containing a large number of classes and functions. While **QtCore** module contains non-GUI functionality for working with file and directory etc., **QtGui** module contains all the graphical controls. In addition, there are modules for working with XML (**QtXml**), SVG (**QtSvg**), and SQL (**QtSql**), etc.

A list of frequently used modules is given below –

- **QtCore** – Core non-GUI classes used by other modules
- **QtGui** – Graphical user interface components
- **QtMultimedia** – Classes for low-level multimedia programming
- **QtNetwork** – Classes for network programming
- **QtOpenGL** – OpenGL support classes
- **QtScript** – Classes for evaluating Qt Scripts
- **QtSql** – Classes for database integration using SQL
- **QtSvg** – Classes for displaying the contents of SVG files
- **QtWebKit** – Classes for rendering and editing HTML
- **QtXml** – Classes for handling XML
- **QtWidgets** – Classes for creating classic desktop-style UIs
- **QtDesigner** – Classes for extending Qt Designer

Supporting Environments

PyQt is compatible with all the popular operating systems including Windows, Linux, and Mac OS. It is dual licensed, available under GPL as well as commercial license. The latest stable version is PyQt5-5.13.2.

Windows

Wheels for 32-bit or 64-bit architecture are provided that are compatible with Python version 3.5 or later. The recommended way to install is using PIP utility –

```
pip3 install PyQt5
```

To install development tools such as Qt Designer to support PyQt5 wheels, following is the command –

```
pip3 install pyqt5-tools
```

You can also build PyQt5 on Linux/macOS from the source code www.riverbankcomputing.com/static/Downloads/PyQt5



Fig no 6.1 QMainWindow

The PyQt installer comes with a GUI builder tool called **Qt Designer**. Using its simple drag and drop interface, a GUI interface can be quickly built without having to write the code. It is however, not an IDE such as Visual Studio. Hence, Qt Designer does not have the facility to debug and build the application.

Start Qt Designer application which is a part of development tools and installed in scripts folder of the virtual environment.

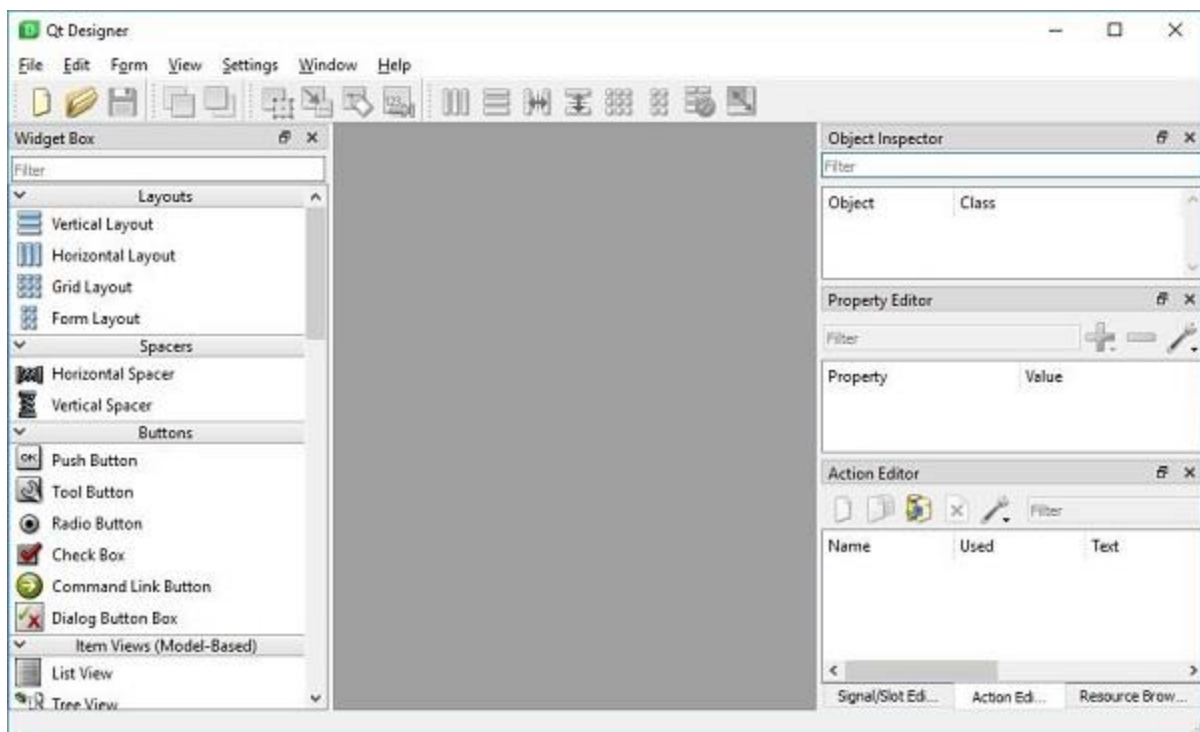


Fig no 6.2 PYQT5 main window

Start designing GUI interface by choosing File → new menu.

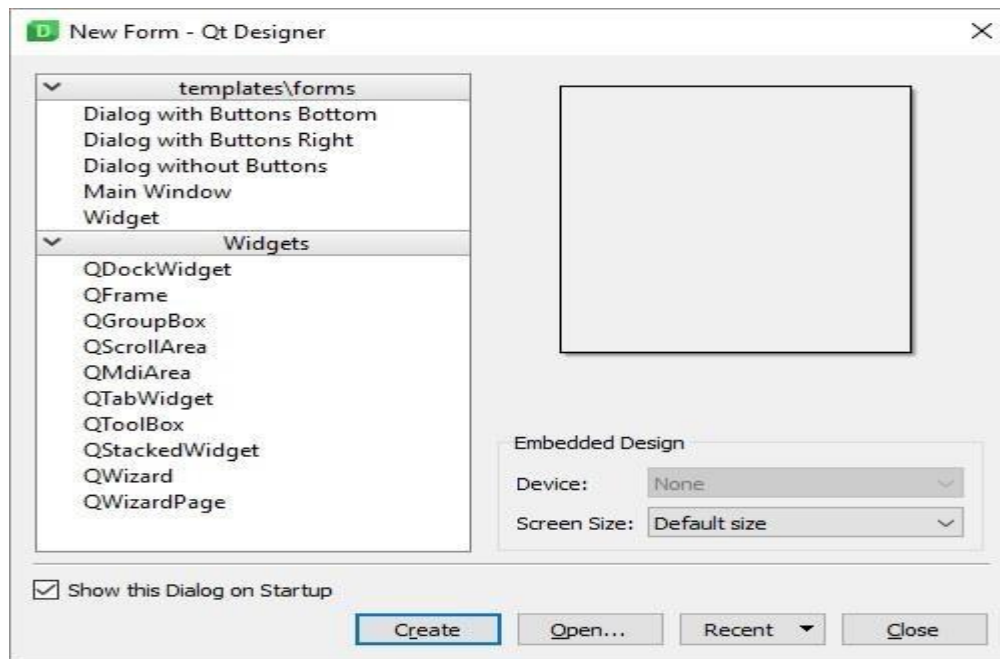


Fig no 6.3 dialog box

You can then drag and drop required widgets from the widget box on the left pane.

You can also assign value to properties of widget laid on the form.

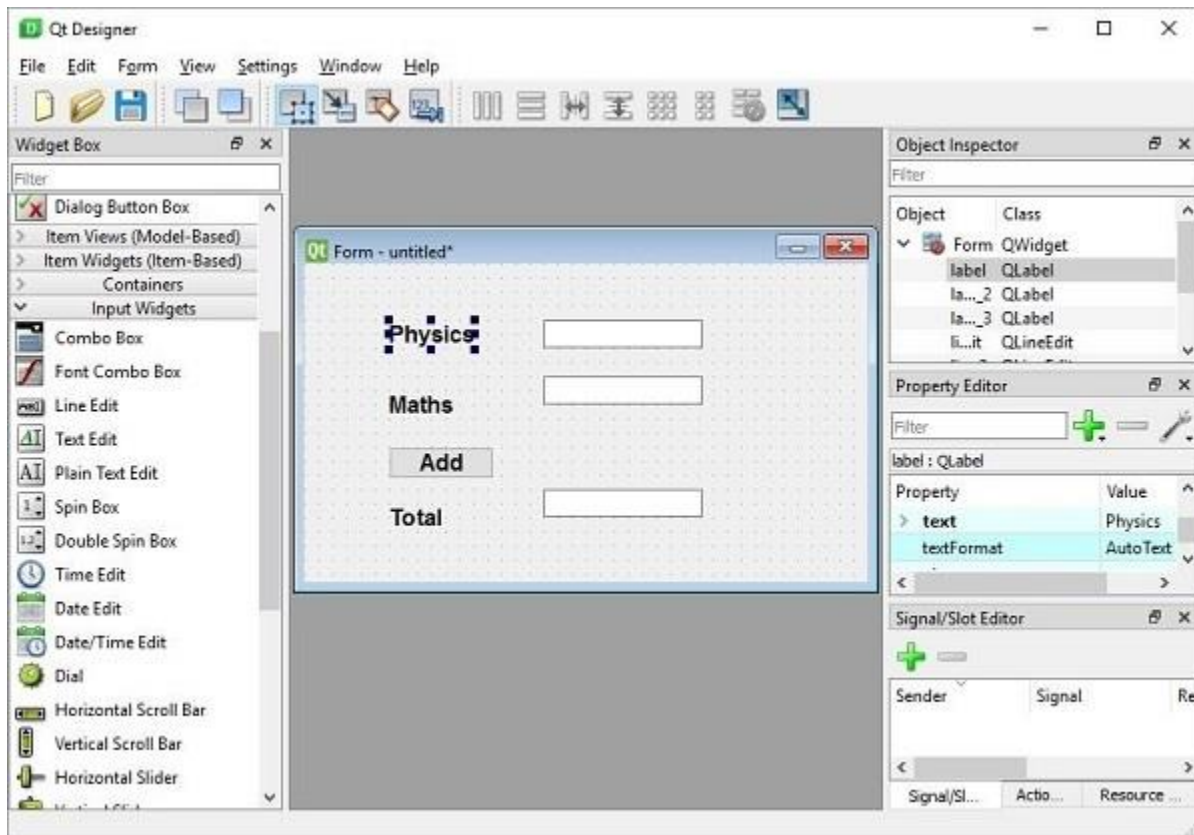


Fig no 6.4 form page

The designed form is saved as demo.ui. This ui file contains XML representation of widgets and their properties in the design. This design is translated into Python equivalent by using pyuic5 command line utility. This utility is a wrapper for uic module of Qt toolkit. The usage of pyuic5 is as follows –

```
pyuic5 -x demo.ui -o demo.py
```

In the above command, -x switch adds a small amount of additional code to the generated Python script (from XML) so that it becomes a self-executable standalone application.

```
if __name__ == "__main__":
```

```
    Import sys
```

```
App = QtGui.QApplication(sys.argv)
```

```
Dialog = QtGui.QDialog()
```

```
ui = Ui_Dialog()
```

```
ui.setupUi(Dialog)
```

```
Dialog.show()
```

```
sys.exit(app.exec_())
```

The resultant python script is executed to show the following dialog box –

Python demo.py

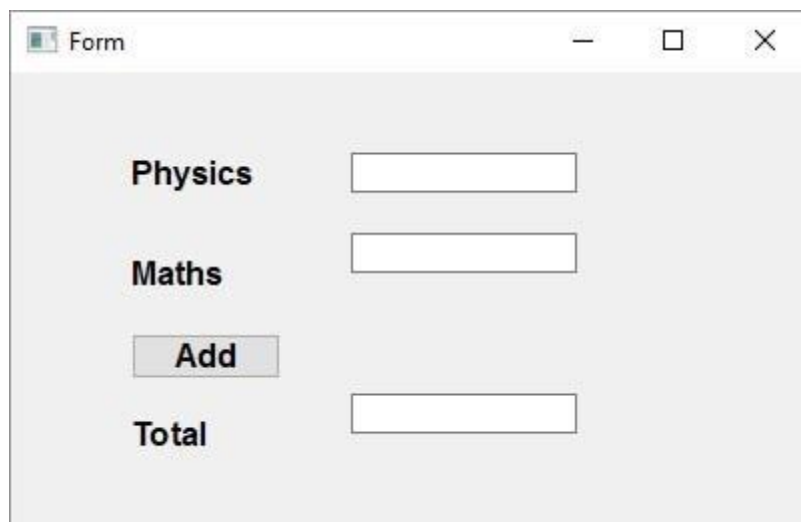


Fig no 6.5 form window

The user can input data in input fields but clicking on Add button will not generate any action as it is not associated with any function. Reacting to user-generated response is called as **event handling**.

Unlike a console mode application, which is executed in a sequential manner, a GUI based application is event driven. Functions or methods are executed in response to

user's actions like clicking on a button, selecting an item from a collection or a mouse click etc., called **events**.

Widgets used to build the GUI interface act as the source of such events. Each PyQt widget, which is derived from QObject class, is designed to emit '**signal**' in response to one or more events. The signal on its own does not perform any action. Instead, it is 'connected' to a '**slot**'. The slot can be any **callable Python function**.

Using Qt Designer's Signal/Slot Editor

First design a simple form with a QLineEdit control and a PushButton.

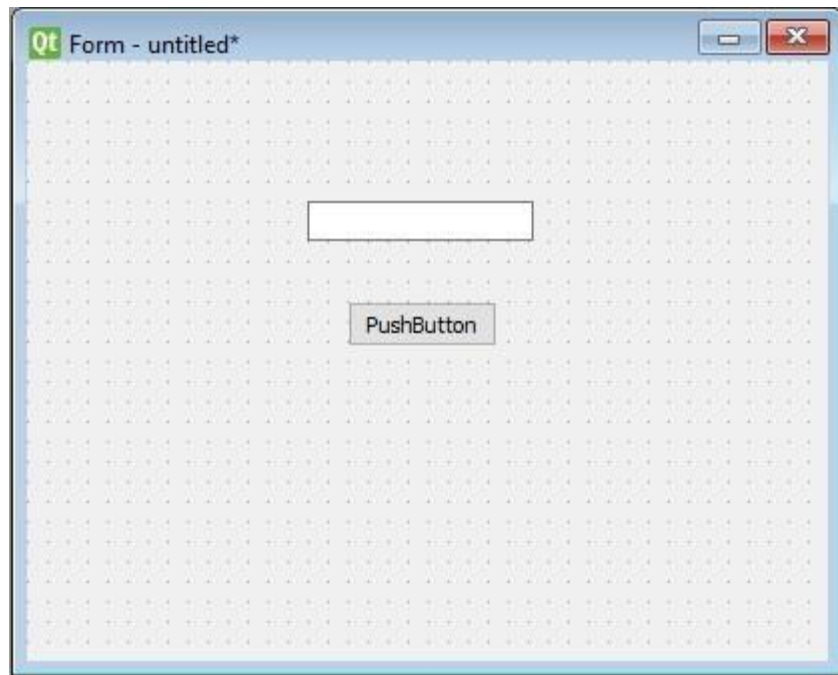


Fig no 6.6 design page

It is desired that if button is pressed, contents of text box should be erased. The QLineEdit widget has a clear () method for this purpose. Hence, the button's **clicked** signal is to be connected to **clear ()** method of the text box.

To start with, choose Edit signals/slots from Edit menu (or press F4). Then highlight the button with mouse and drag the cursor towards the textbox

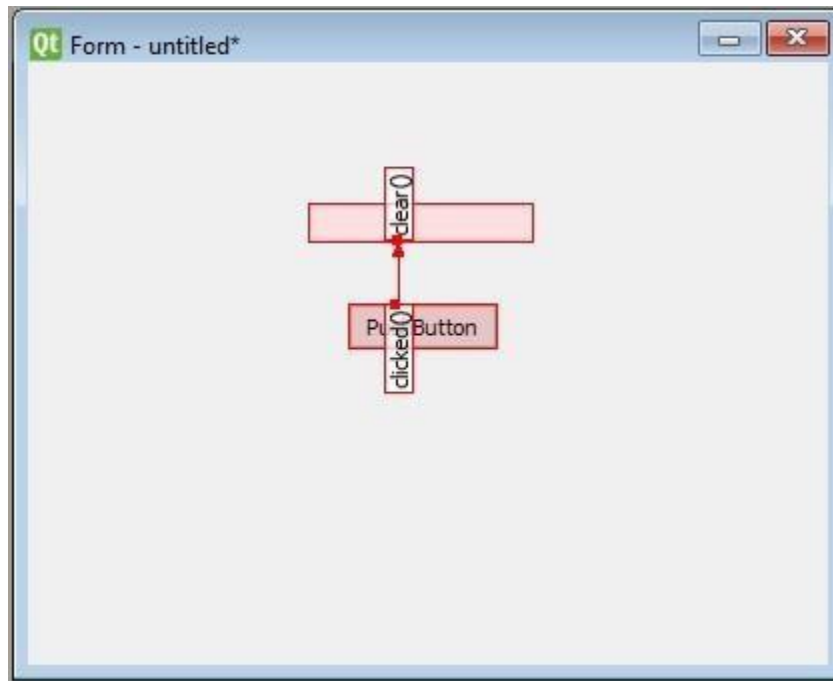


Fig no 6.7 Form design page

As the mouse is released, a dialog showing signals of button and methods of slot will be displayed. Select clicked signal and clear () method

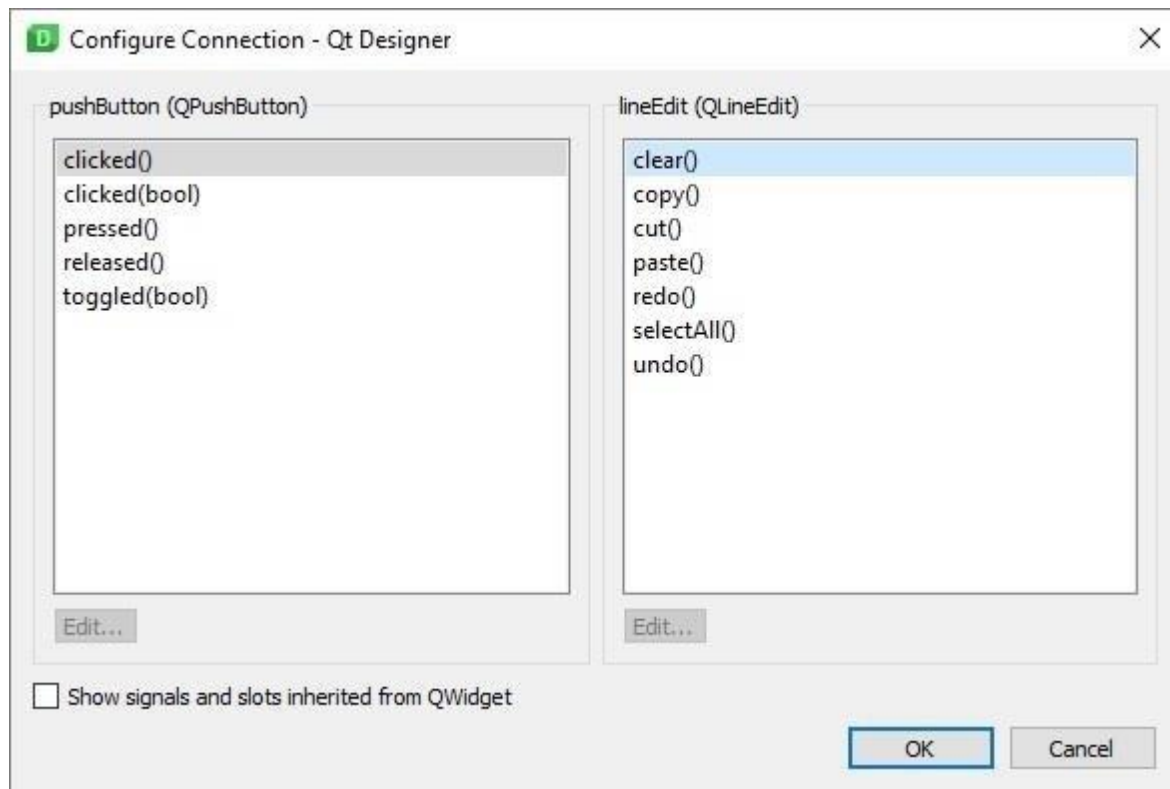


Fig no 6.8 editing page

CHAPTER 7

CONCLUSION

In this project, we have combined cloud technology learning approaches with API tracking system. We have used python coding with cloud technology. We selected pyqt5 for the front end development. Through these it will be very easy to choose topics of news according to our preferences. The repeated news would be ignored giving a quick go through of informations. In future we would be applying system for sports domain, politics, entertainment, science and discovery, etc., Thus getting information in no time is possible through this project.

CHAPTER 9

REFERENCE

1. Kumar: Text Data Pre-processing and Dimensionality Reduction Techniques for Document Clustering Sri Sivani College of Engineering Sri Sivani College of Engineering, vol. 1, no. 5, pp. 1–6 (2019)
2. Saha, Ankan, and Vikas Sindhwani: Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In Proceedings of the fifth ACM international conference on Web search and data mining, pp. 693-702. ACM (2020)
3. Acun, A. Ba, O. Ekin, M. İ. Saraç, and F. Can: Topic Tracking Using Chronological Term Ranking, vol. 25 (2020)
4. Pouliquen, R. Steinberger, C. Ignat, E. Käsper, and I. Temnikova: Multilingual and cross-lingual news topic tracking (2019)
5. Elkan: Text mining and topic models the multinomial distribution (2020)
6. Aksoy, F. Can, and S. Kocberber: Novelty Detection for Topic Tracking, vol. 63, no. 4, pp. 777–795 (2020)
7. Juha Makkonen : Semantic Classes in Topic Detection and Tracking (2018)
8. Cieri, D. Graff, M. Liberman, N. Martey, and S. Strassel: Large , Multilingual , Broadcast News Corpora For Cooperative Research in Topic Detection And Tracking : The TDT-2 and TDT-3 Corpus Efforts, no. January 2018 (2018)
9. Eichmann, David, Miguel Ruiz, Padmini Srinivasan, Nick Street, Chris Culy, and Filippo Menczer: A cluster-based approach to tracking, detection and

segmentation of broadcast news. In Proceedings of the DARPA Broadcast News Workshop, pp. 69-76. (2019).

10. Perez-Tellez, Fernando, David Pinto, John Cardiff, and Paolo Rosso: Clustering weblogs on the basis of a topic detection method. In Mexican Conference on Pattern Recognition, pp. 342-351. Springer Berlin Heidelberg (2021)

11. Fukumoto and Y. Yamaji: LNAI 3651 - Topic Tracking Based on Linguistic Features, pp. 10–21 (2021).

12. I. De: Experiments in First Story Detection, pp. 1–8 (2020)

13. Bigi, Brigitte, Armelle Brun, Jean-Paul Haton, Kamel Smaïli, and Imed Zitouni: Dynamic topic identification: Towards combination of methods (2021).

14. J. Allan, S. Harding, D. Fisher, A. Bolivar, S. Guzman-lara, and P. Amstutz,: Taking Topic Detection From Evaluation to Practice, pp. 1–10 (2019).

15. Schultz, J. Michael, and Mark Liberman: Topic detection and tracking using idf-weighted cosine coefficient. In Proceedings of the DARPA broadcast news workshop, pp. 189-192. San Francisco: Morgan Kaufmann (2018).

16. K. Kaur: International Journal of Advanced Research in A Survey of Topic Tracking Techniques, vol. 2, no. 5, pp. 384–393 (2020)

17. K. Kaur and V. Gupta: Tracking for punjabi language, vol. 1, no. 3, pp. 37– 49 (2019)

18. Hoogma, Niek: The modules and methods of topic detection and tracking. In 2nd Twente Student Conference on IT (2020)

19. M. Mohd: Design and Evaluation of an Interactive Topic Detection and Tracking Interface (2020).
20. Makkonen, Juha: Semantic classes in topic detection and tracking (2021)
21. Bijal, Dalwadi, and Suthar Sanket: Overview of stemming algorithms for Indian and Non-Indian languages, arXiv preprint arXiv: 1404.2878 (2014).
22. Sagar M. Patel, Vipul K. Dabhi, Harshadkumar B. Prajapati: Extractive Based Automatic Text Summarization," Journal of Computers vol. 12, no. 6, pp. 550-563 (2021).