

Industry Oriented Mini Project Report

On

GRIET ASSISTANT

Submitted in partial fulfillment of the requirements for the award of Degree of

Bachelor of Technology

In

Computer Science and Engineering

By

P. SHRAVANI (17241A05G6)

M. ASHWINI (17241A05F5)

E. SRAVANI (18245A0533)

Under the Guidance of

Mrs. D. Sree Vidya

Assistant Professor



Department of Computer Science and Engineering

GOKARAJU RANGARAJU

INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Bachupally, Kukatpally, Hyderabad-500090.

2019-2020



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the mini project dissertation entitled “**GRIET ASSISTANT**” that is being submitted by

P. SHRAVANI (17241A05G6)

M. ASHWINI (17241A05F5)

E. SRAVANI (18245A0533)

in partial fulfillment of the requirement for the award of the degree in **Bachelor of Technology** in Computer Science and Engineering during the academic year 2019-2020.

Mrs.D.Sree Vidya
Guide

Dr.K.Madhavi
Head of Department

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

There are many people who helped me directly and indirectly to complete my project successfully. We would like to take this opportunity to thank one and all.

First of all, we would like to express my deep gratitude towards my internal guide, **Mrs.D.Sree Vidya, Assistant Professor**, for his for their help and constructive criticism in the completion of my dissertation.

We thank our mini Project coordinator **Dr. G. R. Sakthidharan** for his extensive Support. We would like to thank our faculty Coordinators **Mr. V. Srinivas** and **Dr.G. R. Sakthidharan** for giving consistent suggestions and inputs during the project period.

We bestow our sincere thanks to our HOD,**Dr. K. Madhavi, Professor** for providing ample environment and exceptional encouragement to complete this project. We thank our Principal **Dr. J. Praveen** for providing the internal resources and facilities to complete the dissertation.

We encompass our thanks to our Director **Dr. Jandhyala N Murthy** forgiving us constructive inputs and valuable directions for completing our miniproject.

Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

P. SHRAVANI

(17241A05G6)

M. ASHWINI

(17241A05F5)

E. SRAVANI

(18245A0533)



DECLARATION

We hereby declare that the project work titled "**GRIET ASSISTANT**" is the work done during the period from **09-12-2019 to 04-04-2020** and is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** from **Gokaraju Rangaraju Institute of Engineering and Technology**. The result embodied in this project have not been submitted to any other university or Institute for the award any degree or diploma.

P. SHRAVANI
(17241A05G6)

M. ASHWINI
(17241A05F5)

E. SRAVANI
(18245A0533)

ABSTRACT

Google Assistant is a virtual assistant developed by Google supported Artificial Intelligence that's available on mobile and smart home devices. Google Assistant is used to search the Internet, schedule events and alarms, adjust hardware settings on the user's device. This Virtual Assistant can now solve user required tasks by engaging in a two-way conversation using Google's Natural Language Processing.

GRIET Assistant is similar to Google Assistant. We generally solve our required queries with the help of Google Assistant. Similarly, minor details of the college can be known using GRIET Assistant. This project is mainly for seldom visitors to the college.

This project is based on NLP (Natural Language Processing). It takes Speech as input and responds back in speech format itself. This can be implemented using Python inbuilt libraries and packages such as SpeechRecognition and PyAudio. A Database needs to be created which contains unchanged data about the college.

Keywords: Virtual Assistant, Python, Voice to text, Text to Voice, Natural Language Processing.

	Page No.
Certificate	i
Acknowledgement	ii
Declaration	iii
Abstract	iv
1. Introduction	1
1.1 Introduction	1
1.2 Domain description	1
1.2.1 Natural Language Processing	2
1.2.2 Algorithms in NLP	2
1.3 Application	7
2. System Analysis	8
2.1 Objective of Project	8
2.2 Problem Statement	8
2.3 Feasibility Study	8
2.4 Hardware/Software Requirements	8
2.5 Schematic Diagram	36
2.5.1 Speech to text conversion	36
2.5.2 Integration of Python and MySQL	36
2.5.3 Text to speech conversion	37
3. Implementation	38
3.1 Architecture	38
3.2 Modules Description	38
3.2.1 Speech-to-text conversion	39
3.2.2 Integration of Python and Database	39
3.2.3 Text-to-speech conversion	40
3.3 Input and output analysis	40
4. Conclusion and Future enhancement	41
4.1 Conclusion	41
4.2 Future enhancement	41
BIBLIOGRAPHY	42

APPENDICES

- i. Plagiarism Report
- ii. Screen shots
- iii. Sample codes

1. INTRODUCTION

1.1 Introduction

This project is to create a virtual assistant for college. Google Assistant is a virtual assistant developed by Google supported AI that's available on mobile and smart home devices. Google Assistant is employed to look the web, schedule events and alarms, adjust hardware settings on the user's device. This Virtual Assistant can now solve user required tasks by engaging during a two-way conversation using Google's Natural Language Processing. Users primarily interact with the Google Assistant through natural voice, though keyboard input is additionally supported.

Amazon Alexa, known simply as Alexa, is a virtual assistant AI technology developed by Amazon, first utilized in the Amazon Echo smart speakers developed by Amazon Lab126. It's capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, like news.^[1] Most devices with Alexa allow users to activate the device employing a wake-word (such as *Alexa or Amazon*).

This “GRIET ASSISTANT” uses a number of the features of both Google Assistant and Amazon Alexa. It uses Google's speech to text and Text to speech conversions by importing speech recognition and gTTS libraries in Python for Natural Language Processing. As Alexa have a wake word our GRIET ASSISTANT also features a wake word.

This virtual assistant is for seldom visitors to the college. Almost like Google Assistant, if we ask an issue i.e in speech format about the college then this virtual assistant responds back with the acceptable answer in speech format itself.

1.2 Domain Description

⁽¹⁾Data Science is that the field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structured and unstructured data. Data Science is that the concept of unifying statistics, data analysis, processing and machine learning. It uses techniques and theories drawn from many fields within the context of mathematics, statistics, computing and knowledge science.

Generally modern organizations work with huge amounts of data. That data is available in various forms including documents, spreadsheets, audio recordings, emails, JSON, then many, many more. One among the foremost common ways in which such data is recorded is via text. That text is typically quite almost like the natural language that we use in our day-to-day life.

1.2.1 Natural Language Processing

⁽²⁾Everything we express (either verbally or in written) carries huge amounts of information. The topic we elect, our tone, our selection of words, everything adds some sort of information which will be interpreted and value extracted from it. In theory, we will understand and even predict human behavior using that information.

Data generated from conversations, declarations or maybe tweets are samples of unstructured data. Unstructured data doesn't fit neatly into the normal row and column structure of relational databases, and represent the overwhelming majority of knowledge available within the actual world. It is messy and hard to manipulate. Nevertheless, because of the advances in disciplines like machine learning an enormous revolution goes on regarding this subject. Nowadays it's not about trying to interpret a text or speech supported its keywords (the quaint mechanical way), but about understanding the meaning behind those words (the cognitive way). This way it's possible to detect figures of speech like irony, or maybe perform sentiment analysis.

Natural Language Processing or NLP may be a field of AI that provides the machines the power to read, understand and derive meaning from human languages. It is a discipline that focuses on the interaction between data science and human language, and is scaling to a lot of industries. Today NLP is booming by making huge improvements in the access to data and the increase in computational power, which are allowing practitioners to achieve meaningful results in areas like healthcare, media, finance and human resources, among others.

1.2.2 Algorithms in NLP

Bag of words

Bag of words may be a commonly used model that permits you to count all words during a piece of text. Basically it creates an event matrix for the sentence or document, disregarding grammar and ordering . These word frequencies or occurrences are then used as features for training a classifier. To bring a brief example I took the primary sentence of the song “Across the Universe” from The Beatles:

“Words are flowing out like endless rain into a Dixie cup ,They slither while they pass, they slip away across the universe”

Now let's count the words:

Fig 1: Bag of words

	words	rain	a	paper	they	slip	the	universe	...
<i>Words are flowing out like endless rain into a paper cup,</i>	1	1	1	1	0	0	0	0	...
<i>They slither while they pass, they slip away across the universe</i>	0	0	0	0	3	1	1	1	...

This approach may reflect several downsides just like the absence of semantic meaning and context, and therefore the facts that stop words (like “the” or “a”) add noise to the analysis and a few words aren’t weighted accordingly (“universe” weights but the word “they”).

To solve this problem, one approach is to rescale the frequency of words by how often they seem altogether texts (not just the one we are analyzing) in order that the scores for frequent words like “the”, that also are frequent across other texts, get penalized. This approach to scoring is named “Term Frequency Inverse Document Frequency” (TFIDF), and improves the bag of words by weights. Through TFIDF frequent terms within the text are “rewarded” (like the word “they” in our example), but they also get “punished” if those terms are frequent in other texts we include within the algorithm too. On the contrary, this method highlights and “rewards” unique or rare terms considering all texts. Nevertheless, this approach still has no context and semantics.

Tokenization

Tokenization is that the process of segmenting running text into sentences and words. In essence, it’s the task of cutting a text into pieces called tokens, and at an equivalent time discard certain characters, like punctuation. Following our example, the results of tokenization would be:

Fig 2: Tokenization



Although it's going to seem quite basic during this case and also in languages like English that

separate words by a space (called segmented languages) not all languages behave the same, blank spaces alone are not sufficient enough even for English to perform proper tokenization. Splitting on blank spaces may hack what should be considered as one token, as within the case of certain names (e.g. San Francisco or New York) or borrowed foreign phrases (e.g. laissez faire).

Tokenization can remove punctuation too, easing the path to proper word segmentation but also triggering possible complications. In the case of periods that follow abbreviation (e.g. dr.), the amount following that abbreviation should be considered as part of an equivalent token and not be removed.

The tokenization processes are often particularly problematic when handling biomedical text domains which contain many hyphens, parentheses, and other punctuation marks.

Stop Words removal

Stop word removal includes getting obviate common language articles, pronouns and prepositions like “and”, “the” or “to” in English. during this process some quite common words that appear to supply little or no value to the NLP objective are filtered and excluded from the text to be processed, hence removing widespread and frequent terms that aren't informative about the corresponding text.

Stop words are often safely ignored by completing a lookup during a pre-defined list of keywords, freeing up database space and improving time interval.

There is no universal list of stop words. These are often pre-selected or built from scratch. A possible approach is to start by adopting pre-defined stop words and add words to the list afterward. Nevertheless it seems that the overall trend over the hobby has been to travel from the utilization of huge standard stop word lists to the utilization of no lists in the least.

The thing is stop words removal can wipe out relevant information and modify the context during a given sentence. For instance, if we are performing a sentiment analysis we'd throw our algorithm astray if we remove a stop word like “not”. Under these conditions, you would possibly select a minimal stop glossary and add additional terms counting on your specific objective.

Stemming

Stemming refers to the method of slicing the top or the start of words with the intention of removing affixes (lexical additions to the basis of the word).

Affixes that are attached at the start of the word are called prefixes (e.g. “astro” within the word “astrobiology”) and therefore the ones attached at the top of the word are called suffixes (e.g. “ful” in the word “helpful”).

The problem is that affixes can create or expand new sorts of an equivalent word (called inflectional affixes), or maybe create new words themselves (called derivational affixes). In English, prefixes are always derivational (the affix creates a replacement word as within the example of the prefix “eco” within the word “ecosystem”), but suffixes are often derivational (the affix creates a replacement word as within the example of the suffix “ist” within the word “guitarist”) or inflectional (the affix creates a replacement sort of word as in the example of the suffix “er” within the word “faster”).

Ok, so how can we tell the difference and chop the proper bit?

Fig 3: Stemming



A possible approach is to think about an inventory of common affixes and rules (Python and R languages have different libraries containing affixes and methods) and performs stemming supported them, but of course this approach presents limitations. Since stemmers use algorithmic approaches, the result of the stemming process may not be an actual word or even change the word (and sentence) meaning. To offset this effect you'll edit those predefined methods by adding or removing affixes and rules, but you want to consider that you simply could be improving the performance in one area while producing degradation in another one. Always check out the entire picture and test your model's performance.

So if stemming has serious limitations, why can we use it? First of all, it can be used to correct spelling errors from the tokens. Stemmers are simple to use and run in no time (they perform simple operations on a string), and if speed and

performance are important within the NLP model, then stemming is certainly the way to go. Remember, we use it with the target of improving our performance, not as a grammar exercise.

Lemmatization

Lemmatization has the target of reducing a word to its base form and grouping together different sorts of an equivalent word. For example, verbs in past are become present (e.g. “went” is modified to “go”) and synonyms are unified (e.g. “best” is modified to “good”), hence standardizing words with similar aiming to their root. Although it seems closely associated with the stemming process, lemmatization uses a special approach to succeed in the basis sorts of words. Lemmatization resolves words to their dictionary form (known as lemma) that it requires detailed dictionaries during which the algorithm can check out and link words to their corresponding lemmas.

For example, the words “running”, “runs” and “ran” are all forms of the word “run”, so “run” is the lemma of all the previous words.

Fig 4: Lemmatization



Lemmatization also takes into consideration the context of the word so as to unravel other problems like disambiguation, which suggests it can discriminate between identical words that have different meanings depending on the specific context. Think about words like “bat” (which can correspond to the animal or to the metal/wooden club utilized in baseball) or “bank” (corresponding to the financial organization or to the land alongside a body of water). By providing a part-of-speech parameter to a word (whether it's a noun, a verb, then on) it's possible to define a task for that word within the sentence and remove disambiguation.

Lemmatization may be a far more resource-intensive task than performing a stemming process. At an equivalent time, since it requires more knowledge about the language structure than a stemming approach, it demands more computational power than fixing or adapting a stemmer.

1.3 Application

Gokaraju Rangaraju Institute of Engineering and Technology is an autonomous engineering college in Hyderabad, Telangana, India. It is approved by the “All India Council for Technical Education” and Accredited by National Board of Accreditation (NBA) with an ‘A’ grade.

As we know every year there will be hundreds of new admissions. Freshers visit the college with their parents or family members before their actual joining date. They will have many doubts to be cleared about the college. All of their doubts cannot be cleared at the administration block.

In this situation, our “GRIET Assistant” helps them to clarify their queries about the college.

2. SYSTEM ANALYSIS

2.1 Objective of Project

The objective is to accomplish a model which can engage in a two-way conversation, using Python inbuilt packages and libraries.

2.2 Problem Statement

To create a simple stand-alone application that helps less tech savvy people or seldom visitors of the college to know minor details of the college without feeling ignorant or computer illiterate.

2.3 Feasibility Study

The Feasibility Study examines how well the assistant responds to the required query. As the Database contains only limited amount of the Data, it responds efficiently and correctly only when we ask a particular question for which the answer is present in the database. And yes making the virtual assistant efficient is possible by extending the database with more essential data.

2.4 Hardware/Software requirements

Hardware:

- 2 GB RAM.
- 500 GB Hard Disk.

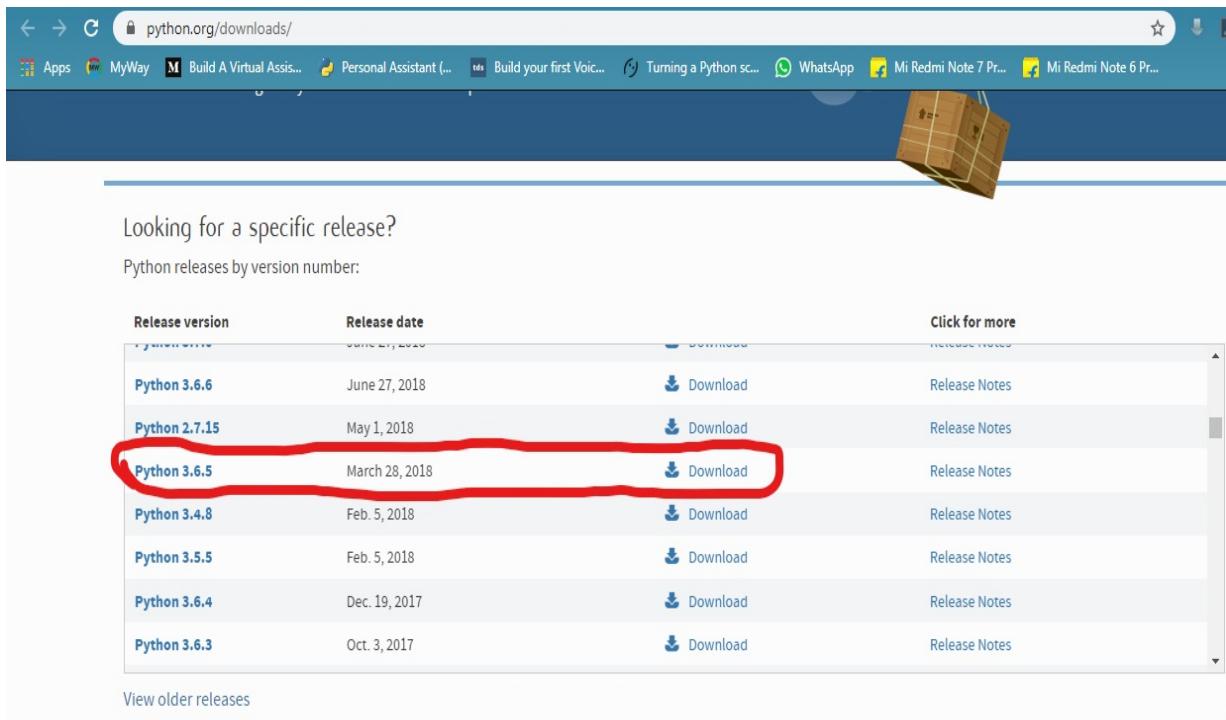
Software:

- OS: Windows 7 and above.
- Python version – 3.6.5
 - Pip version – 9.0.3

As our project is similar to Google assistant, we use Google speech API for our application. Google already have Speech-to-text and Text-to-Speech conversion APIs. There are some of the inbuilt libraries in Python which we can import in our Python code and use those speech conversion APIs.

⁽³⁾Installing Python 3.6.5

Step 1: To download and install Python visit the official website of Python <https://www.Python.org/downloads/> and choose your version. We have chosen Python version 3.6.5

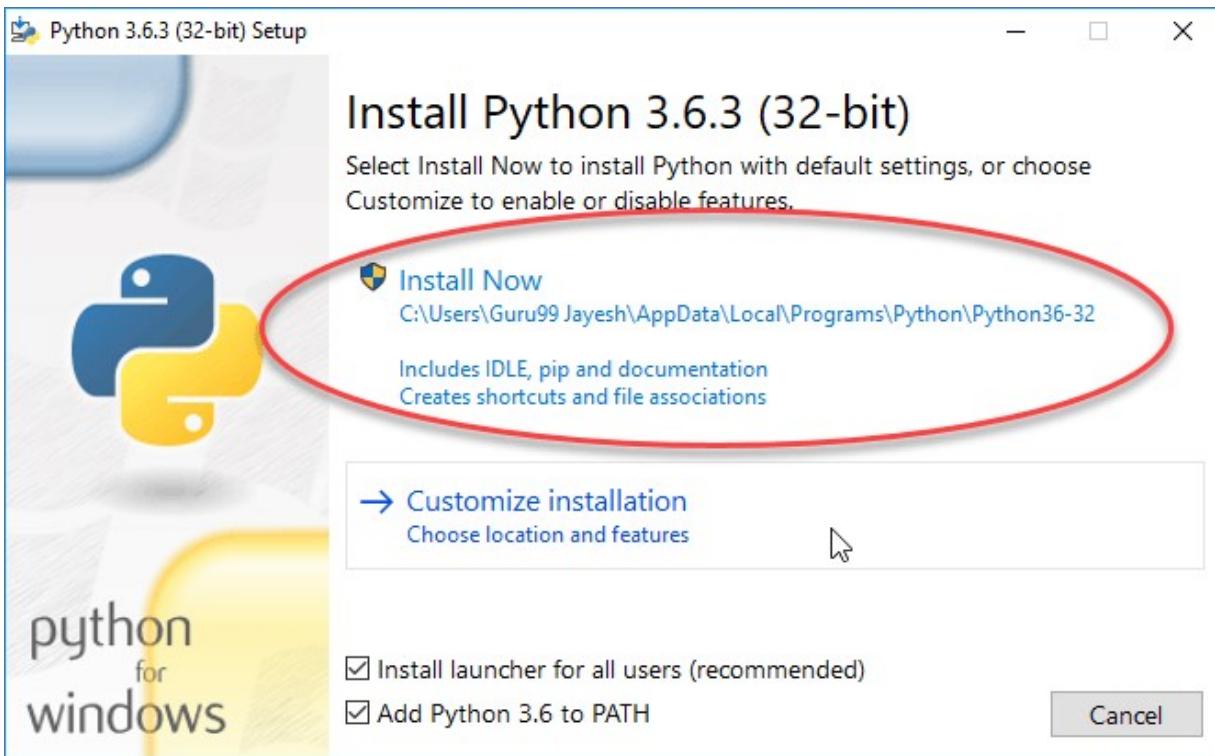


The screenshot shows a web browser displaying the Python.org downloads page. The URL in the address bar is [python.org/downloads/](https://www.Python.org/downloads/). The page header includes a navigation bar with links like Apps, MyWay, Build A Virtual Assis..., Personal Assistant..., Build your first Voic..., Turning a Python sc..., WhatsApp, Mi Redmi Note 7 Pr..., and Mi Redmi Note 6 Pr... There is also a search bar and a login button. Below the header, there's a large image of a brown cardboard box with a yellow ribbon. The main content area has a heading "Looking for a specific release?" followed by "Python releases by version number:". A table lists the following Python releases:

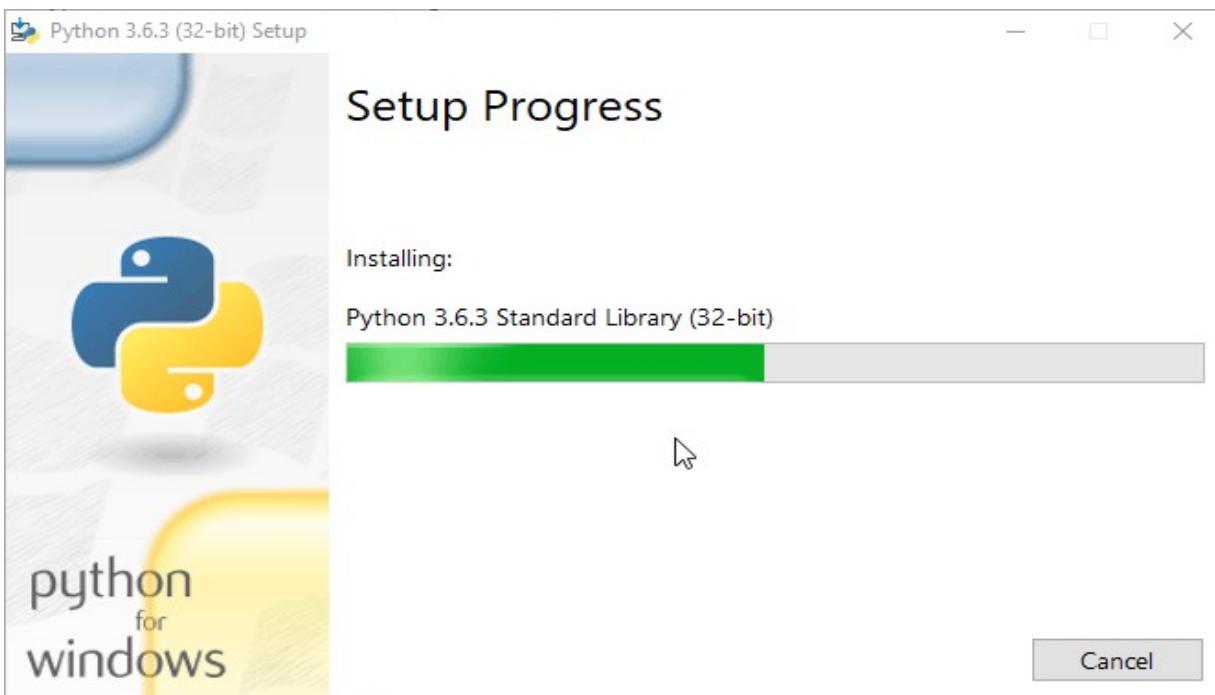
Release version	Release date	Click for more
Python 3.6.6	June 27, 2018	Download Release Notes
Python 2.7.15	May 1, 2018	Download Release Notes
Python 3.6.5	March 28, 2018	Download Release Notes
Python 3.4.8	Feb. 5, 2018	Download Release Notes
Python 3.5.5	Feb. 5, 2018	Download Release Notes
Python 3.6.4	Dec. 19, 2017	Download Release Notes
Python 3.6.3	Oct. 3, 2017	Download Release Notes

A red circle highlights the "Python 3.6.5" row. At the bottom of the table, there is a link "View older releases".

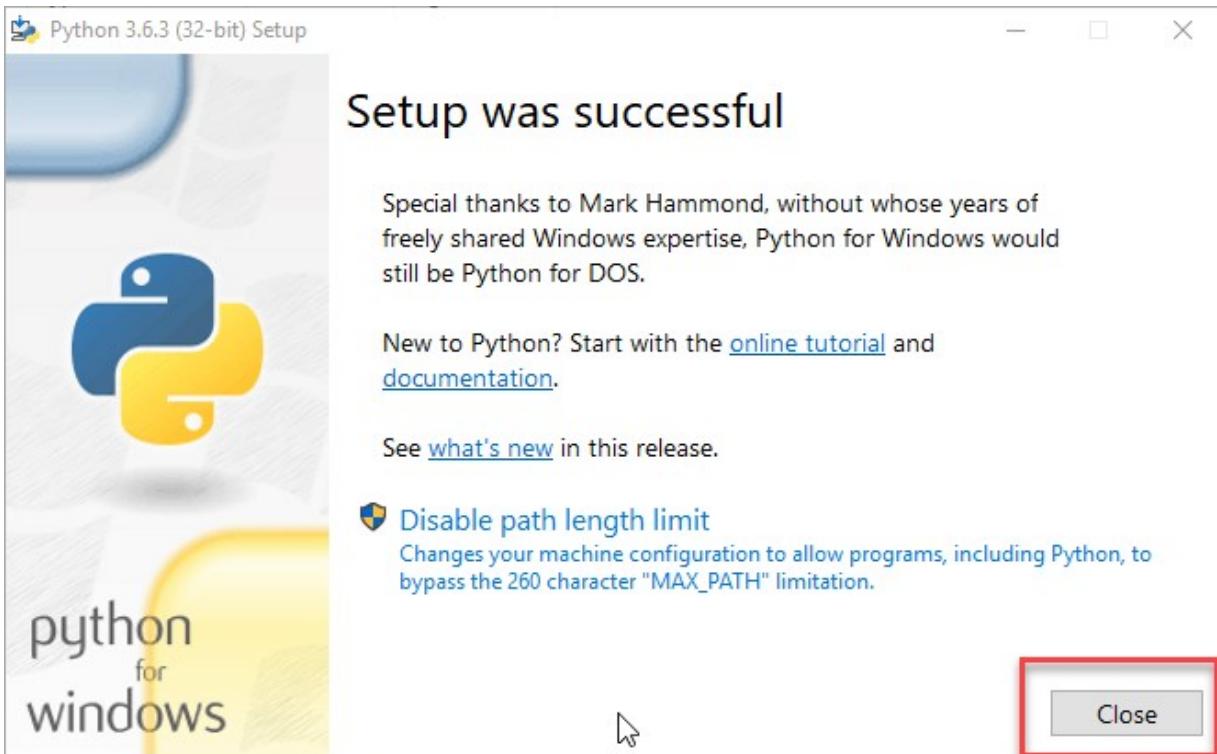
Step 2: Once the download is complete, run the exe for install Python. Now click on Install Now. Make sure that you mark both the check boxes below because it sets its Python path automatically.



Step 3: Now we see Python installing.



Step 4: When it finishes, you'll see a screen that says the Setup was successful. Now click on "Close".



- PyCharm IDE.

⁽⁴⁾**PyCharm** is an integrated development environment (IDE) utilized in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django also as Data Science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there's also Professional Edition with extra features released under a proprietary license.

Features:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages

- Python refactoring: includes rename, extract method, introduce variable, introduce constant, and pull up, push down and others.
- Support for web frameworks: Django, web2py and Flask [professional edition only]
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development [professional edition only]
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
- Support for scientific tools like matplotlib, numpy and scipy [professional edition only]^[10]
- It competes mainly with a number of other Python-oriented IDEs, including Eclipse's PyDev, and the more broadly focused Komodo IDE.

Plugins:

PyCharm provides API in order that developers can write their own plugins to increase PyCharm features. Several plugins from other JetBrains IDE also work with PyCharm. There are quite 1000 plugins which are compatible with PyCharm.

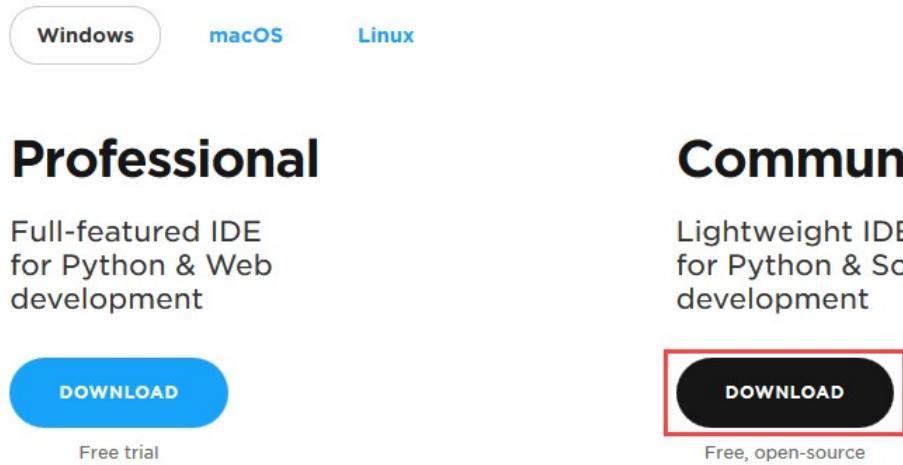
So, to get started we need to install Python and PyCharm

(5) Installing PyCharm

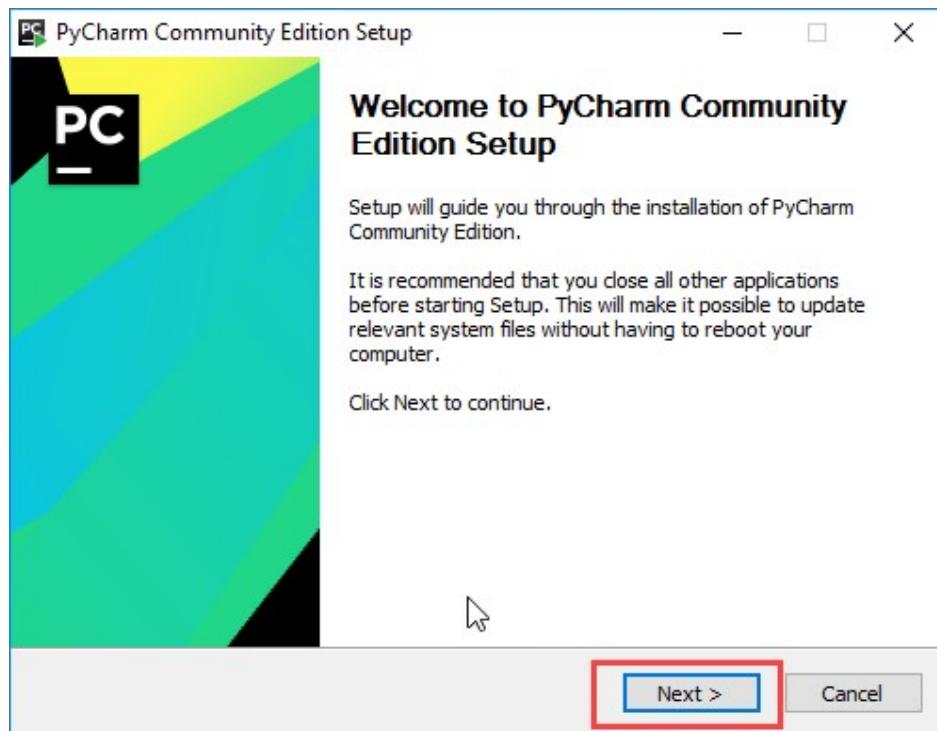
Step 1: To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and Click the "DOWNLOAD"

link under the Community Section

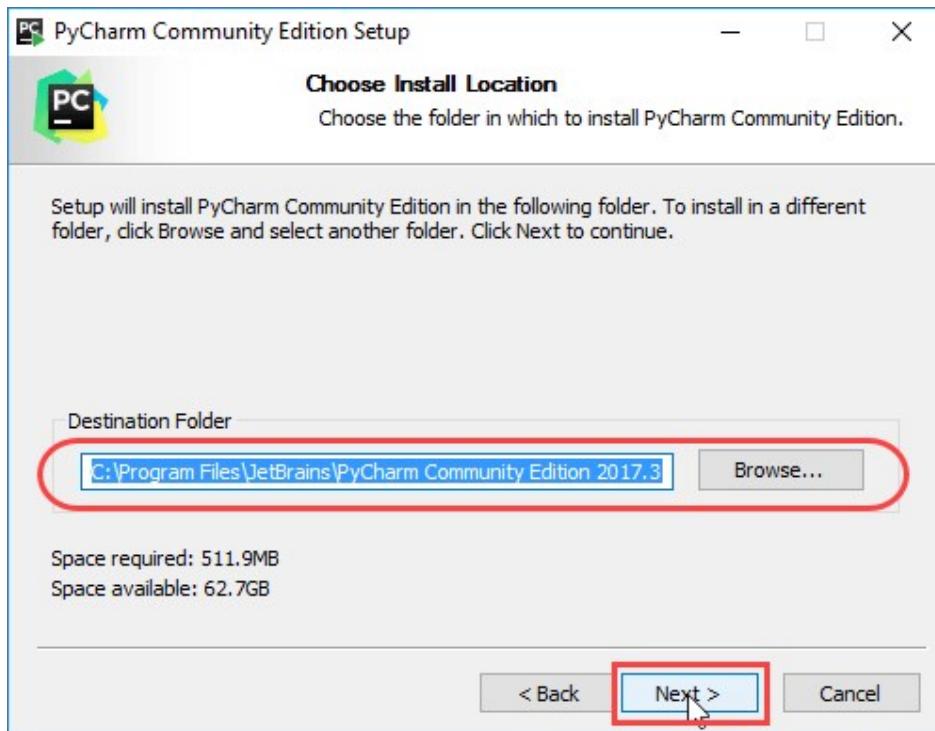
Download PyCharm



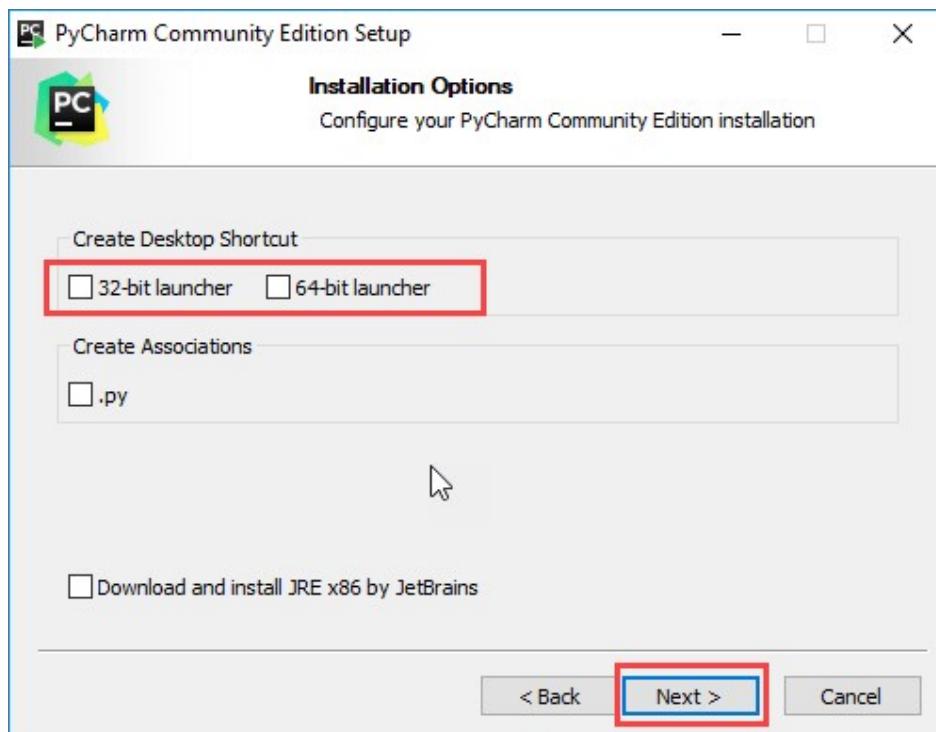
Step 2: Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.



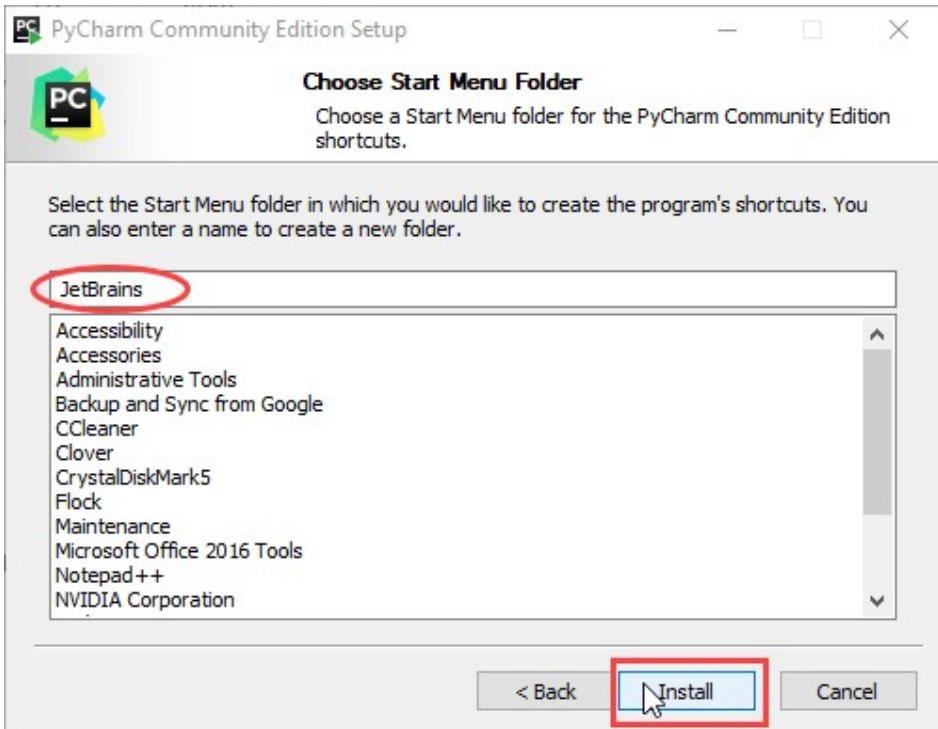
Step 3: On subsequent screen, Change the installation path if required. Click “Next”.



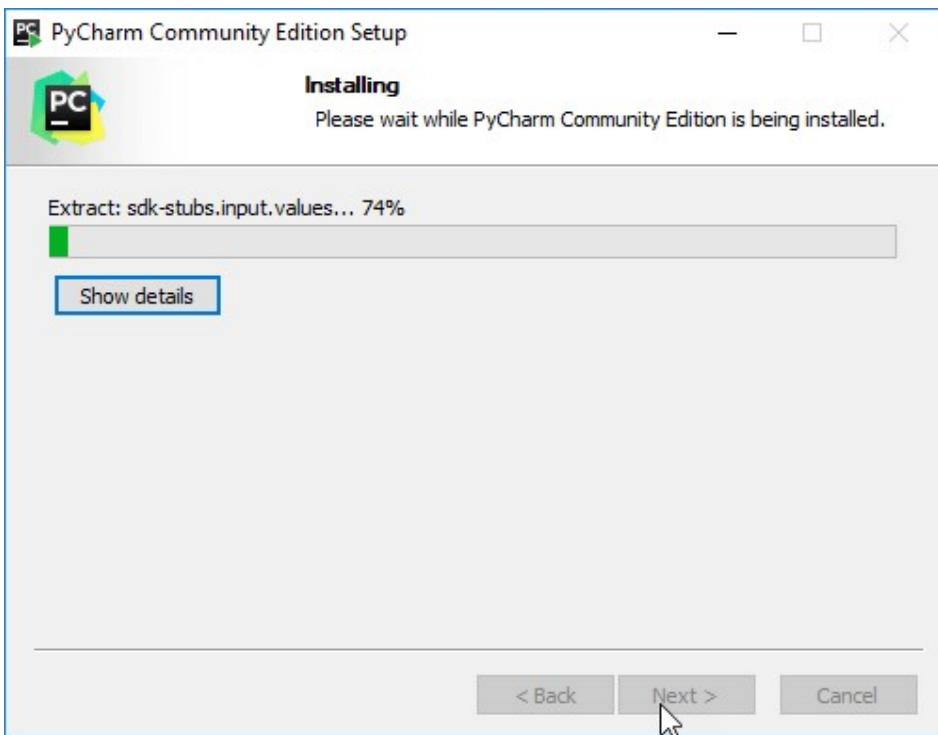
Step 4: On subsequent screen, you'll create a desktop shortcut if you would like and click on “Next”.



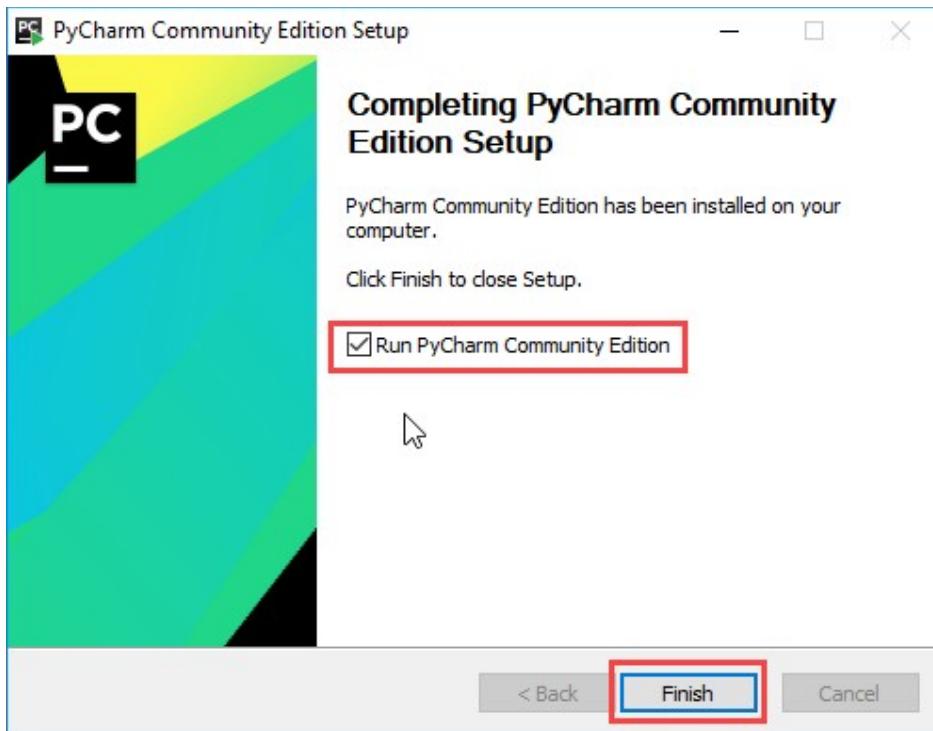
Step 5: Choose the start menu folder. Keep selected JetBrains and click on “Install”.



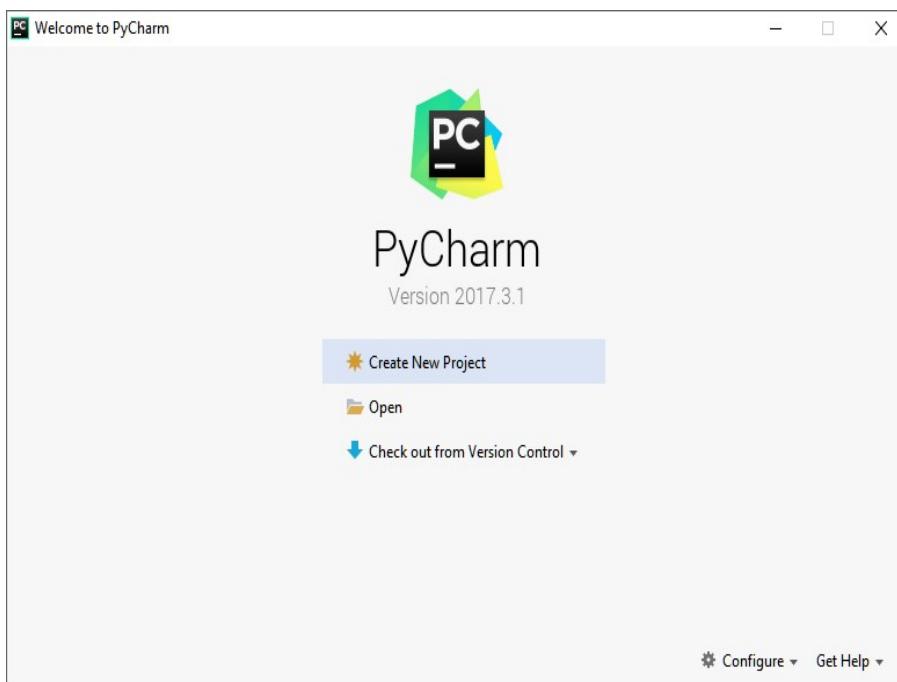
Step 6: Wait for the installation to finish.



Step 7: Once installation finished, you ought to receive a message screen that PyCharm is installed. If you would like to go on and run it, click the “Run PyCharm Community Edition” box first and click on “Finish”.

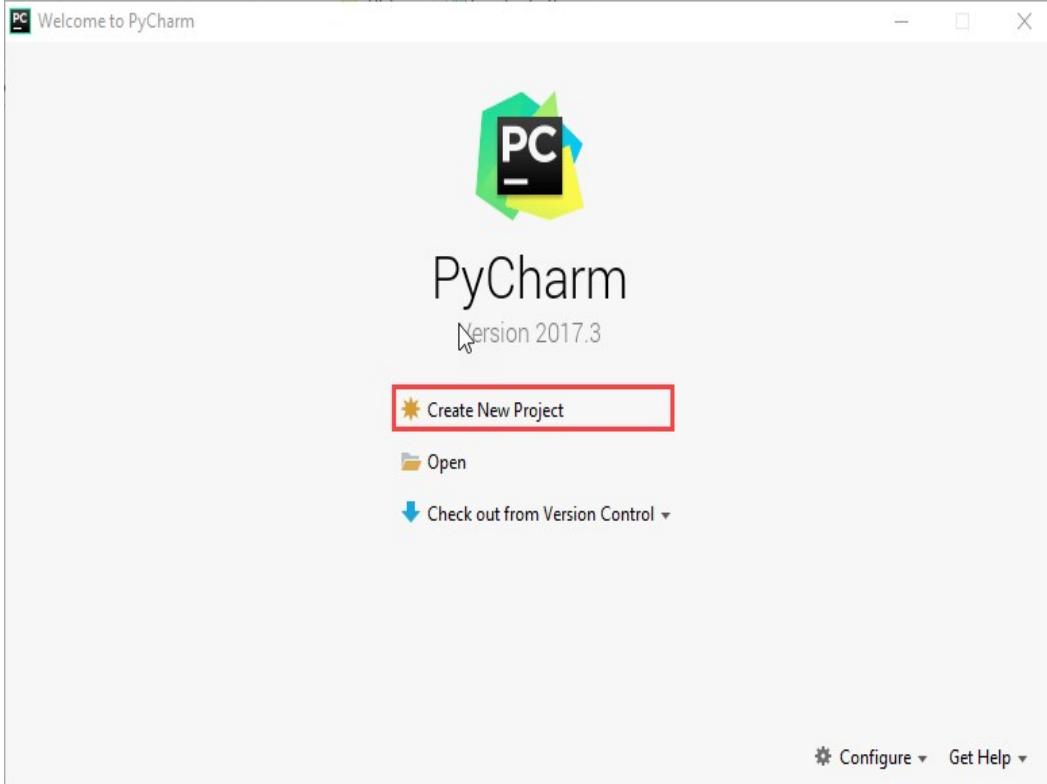


Step 8: After you click on "Finish," the subsequent screen will appear.



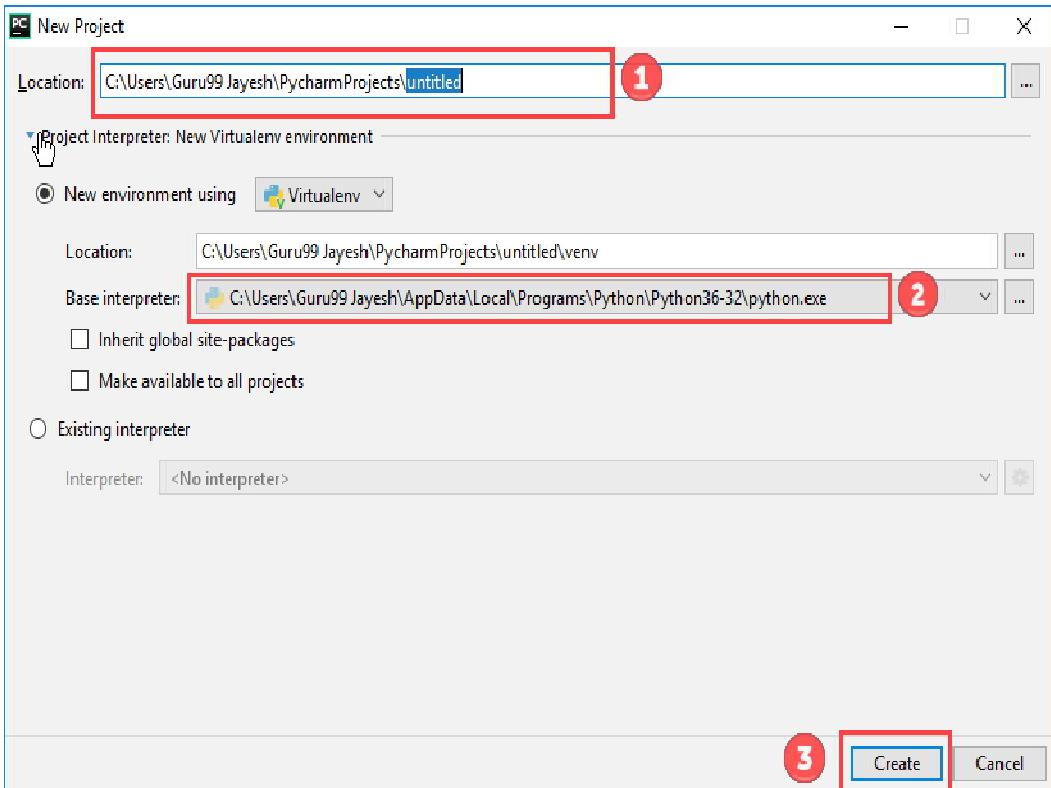
Creating First Program in PyCharm

Step 1: Open PyCharm Editor. You can see the introductory screen for PyCharm. To create a new project, click on “Create New Project”.

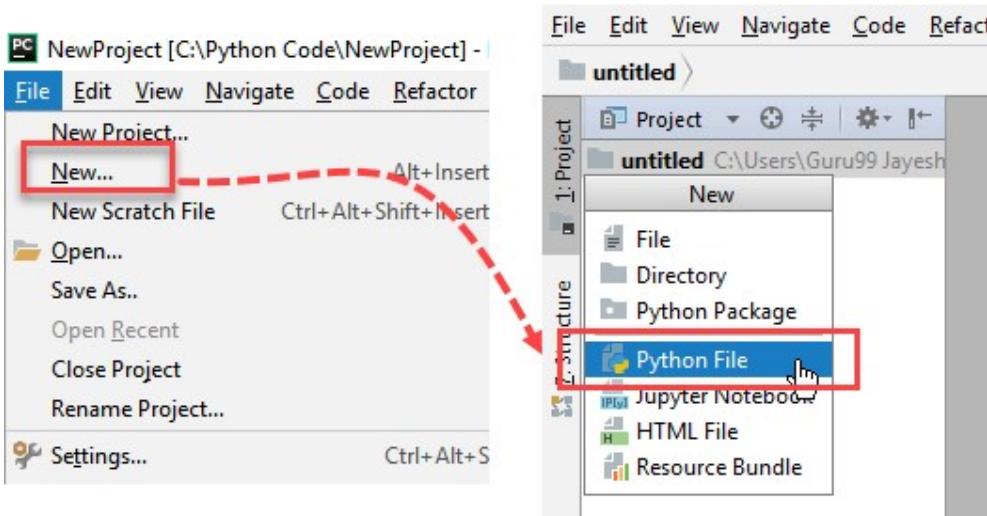


Step 2: You will need to select a location.

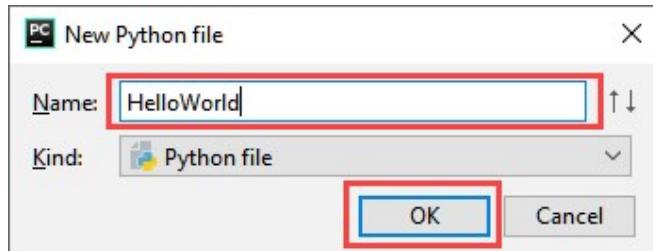
1. You can select the location where you want the project to be created. If you don't want to vary location than keep it as it is but just change the name from "untitled" to something more meaningful, like "FirstProject".
2. PyCharm should have found the Python interpreter you installed earlier.
3. Next Click the "Create" Button.



Step 3: Now Go up to the “File” menu and choose “New”. Next, select “Python File”.



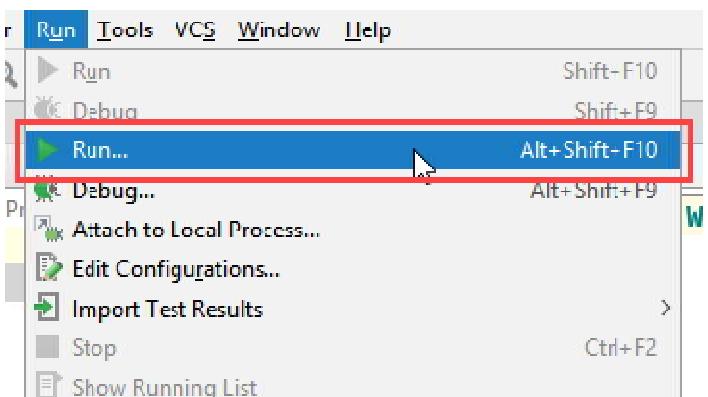
Step 4: A new pop up will appear. Now type the name of the file you would like (Here we give “HelloWorld”) and hit “OK”.



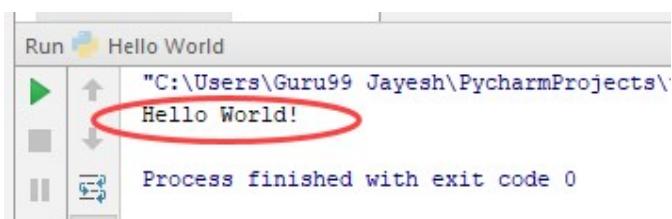
Step 5: Now write a simple program - print ('Hello World!').

```
1 print("Hello World!")  
2  
3  
4
```

Step 6: Now go up to the “Run” menu and choose “Run” to run your program.



Step 7: You'll see the output of your program at rock bottom of the screen.



Step 8: Don't be concerned if you do not have Pycharm Editor installed, you'll still run the code from the command prompt. Enter the correct path of a file in command prompt to run the program.

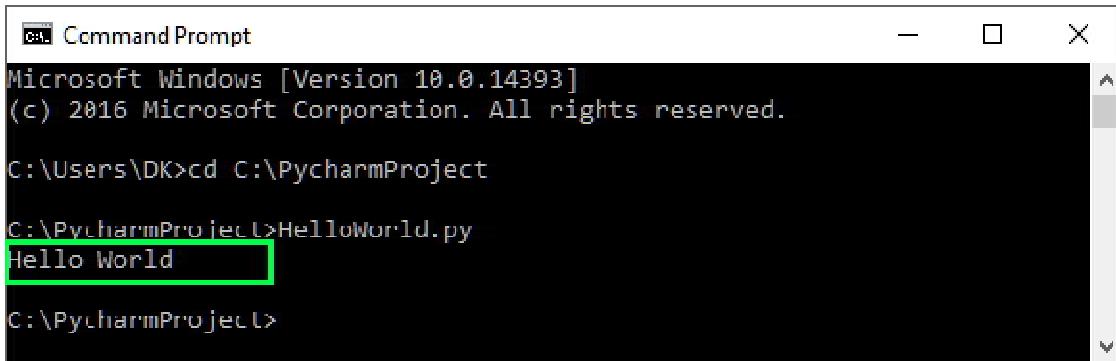


```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\DK>cd C:\PycharmProject
C:\PycharmProject>HelloWorld.py
```

Run your python code from command prompt

The output of the code would be



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\DK>cd C:\PycharmProject
C:\PycharmProject>HelloWorld.py
Hello World
C:\PycharmProject>
```

MySQL Workbench

⁽⁶⁾MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system.

⁽⁷⁾Installation of MySQL Workbench

Step 1: Go to <https://dev.MySQL.com/downloads/workbench/> and click on “Go to download page”.

← → C dev.mysql.com/downloads/workbench/

Apps Building Blocks Indi... Facebook - Log In o... TRACK YOUR PAN... SEARCH

MySQL Workbench 8.0.19

Select Operating System:

Microsoft Windows

Recommended Download:

MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

Go to Download Page >

Other Downloads:

Windows (x86, 64-bit), MSI Installer 8.0.19 35.6M Download

(mysql-workbench-community-8.0.19-winx64.msi) MD5: 3e802457a04c7e58c4abbb914630f510 | Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

Step 2: Click on the Download button which downloads the windows installer file for Windows 32 bit that is “MySQL-installer-community-8.0.19.0.msi”

← → C dev.mysql.com/downloads/windows/installer/8.0.html

Apps Building Blocks Indi... Facebook - Log In o... TRACK YOUR PAN... SEARCH

MySQL Installer

MySQL Installer 8.0.19

Select Operating System:

Microsoft Windows

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer 8.0.19 18.6M Download

(mysql-installer-web-community-8.0.19.0.msi) MD5: 32043776cb2239db45fddaa86dc0ad61 | Signature

Windows (x86, 32-bit), MSI Installer 8.0.19 398.9M Download

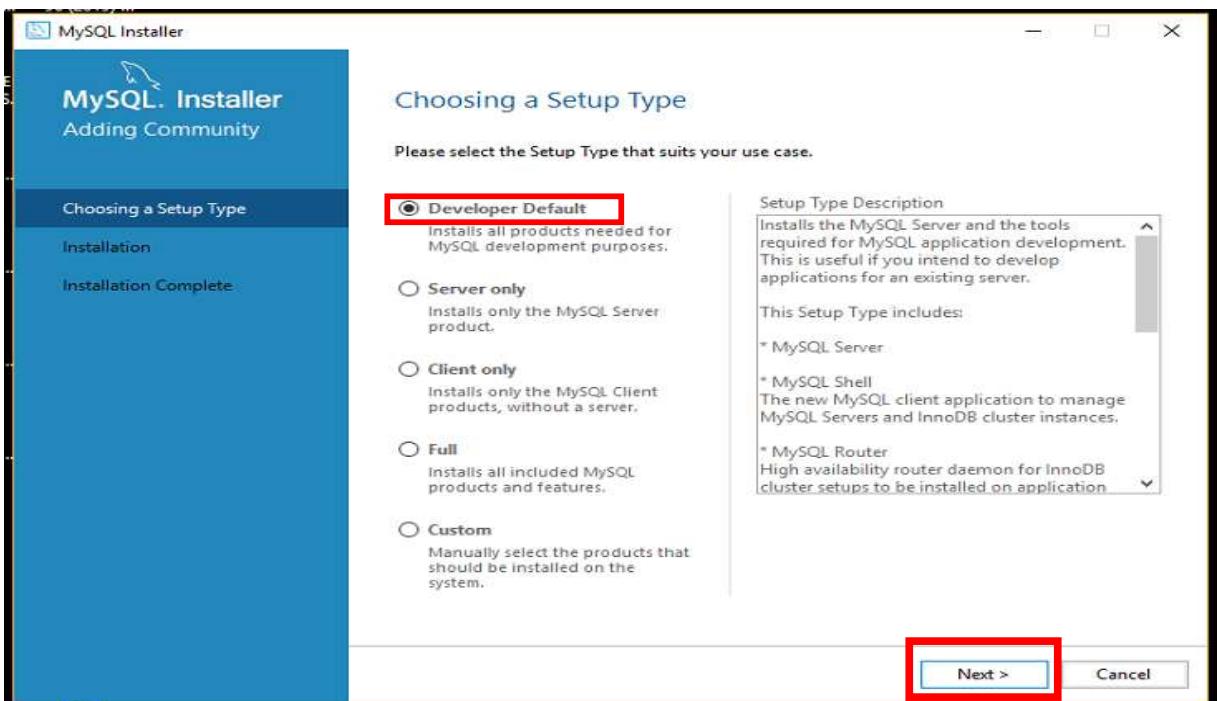
(mysql-installer-community-8.0.19.0.msi) MD5: 1a882015da7fb93f20c4717e63b6817c | Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

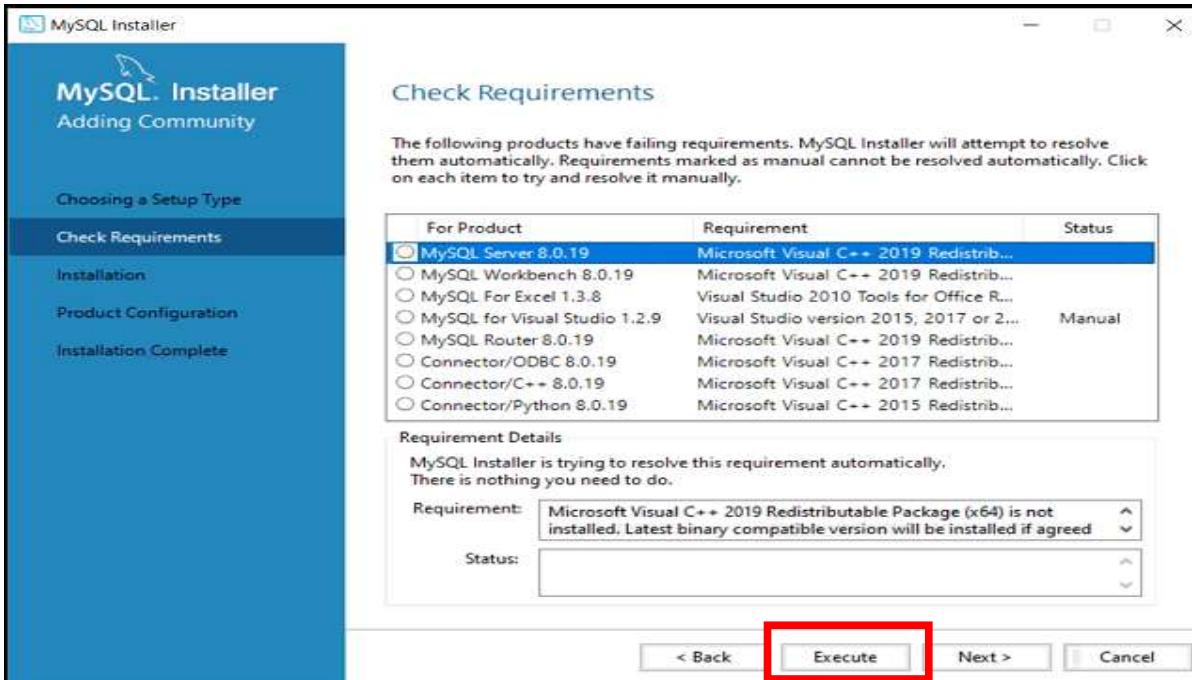
Step 3: It will ask for login or sign up for Oracle account, but no need, we just have to click on “No thanks, just start my download”.

The screenshot shows a web browser window with the URL dev.mysql.com/downloads/file/?id=492815. The page title is "MySQL Community Downloads". It features two buttons: "Login » using my Oracle Web account" and "Sign Up » for an Oracle Web account". Below these buttons, a note states: "MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions." At the bottom left, there is a link "[No thanks, just start my download.](#)". This link is highlighted with a red rectangular box. The address bar at the top shows the full URL: "https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-community-8.0.19.0.msi".

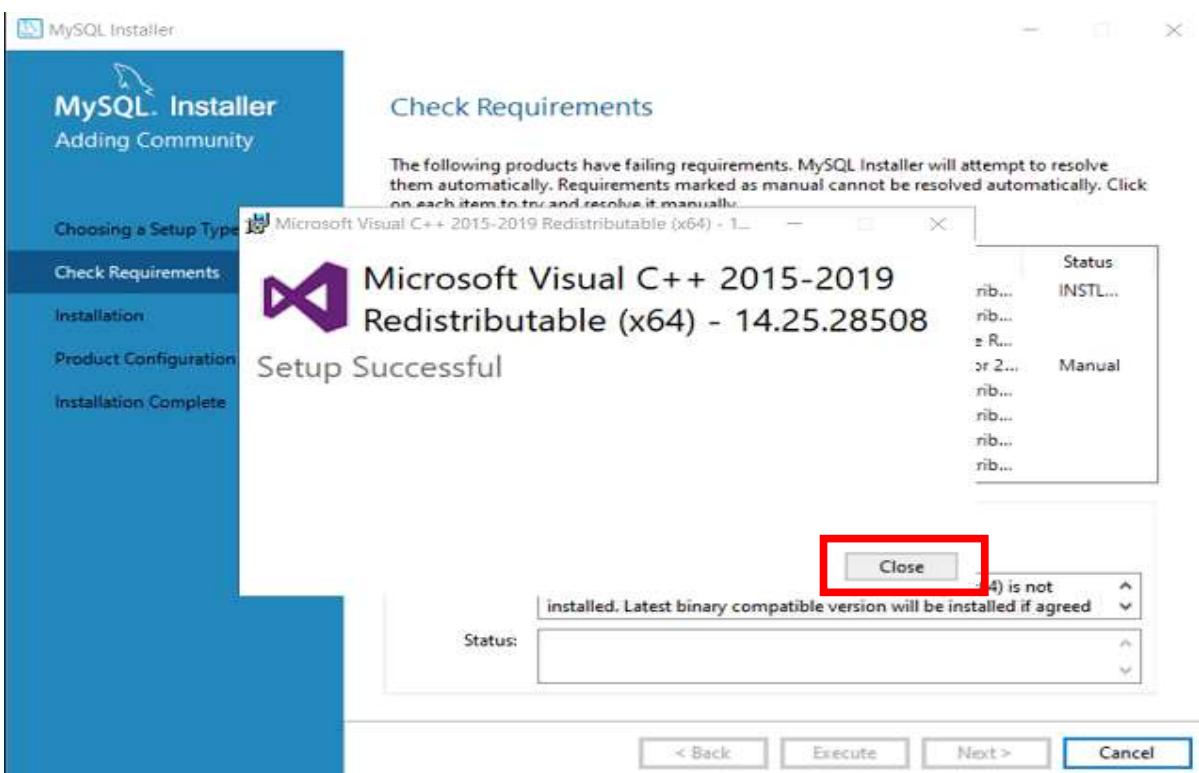
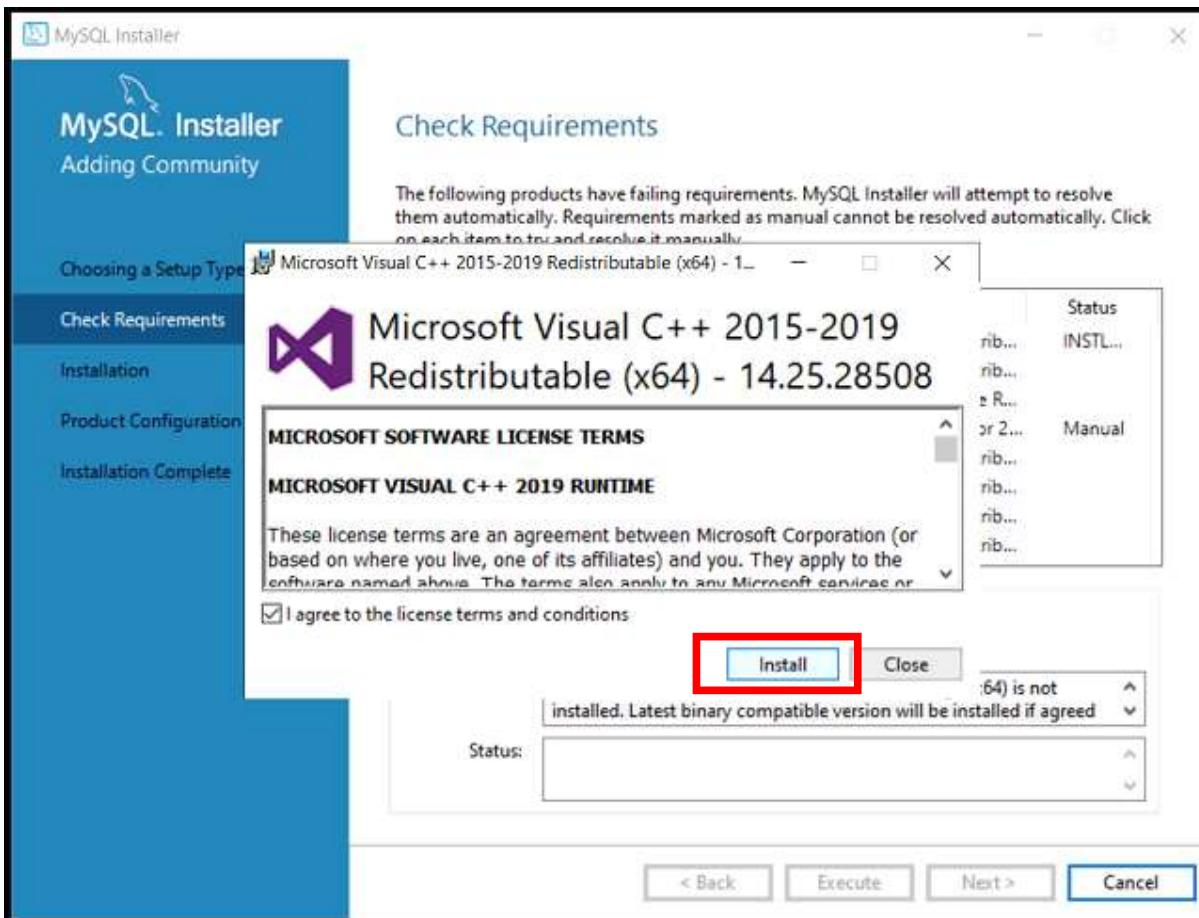
Step 4: Run the windows installer file which got downloaded. After giving all the installation permissions the following window opens, select “Developer Default” and click on next.



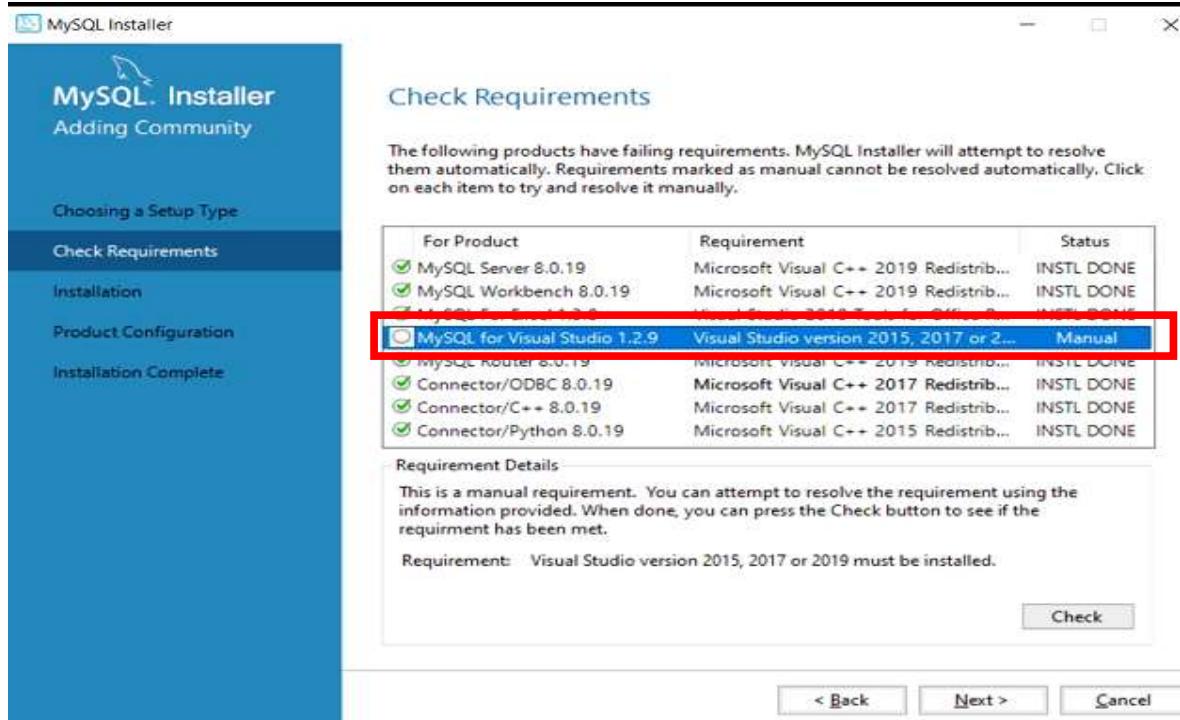
Step 5: Now it checks for some requirements. Click on execute which then installs Microsoft visual C++ 2019.



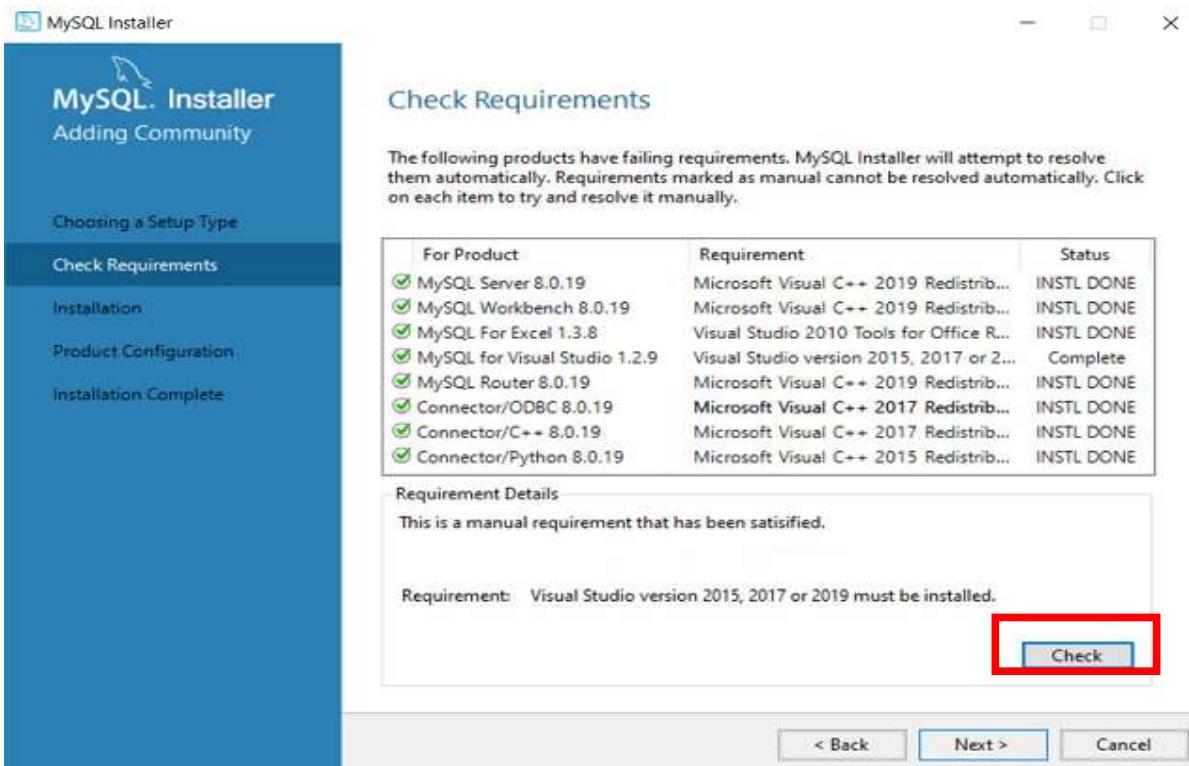
Step 6: Agree to the terms and conditions and click on install for installing Microsoft Visual C++ 2019.



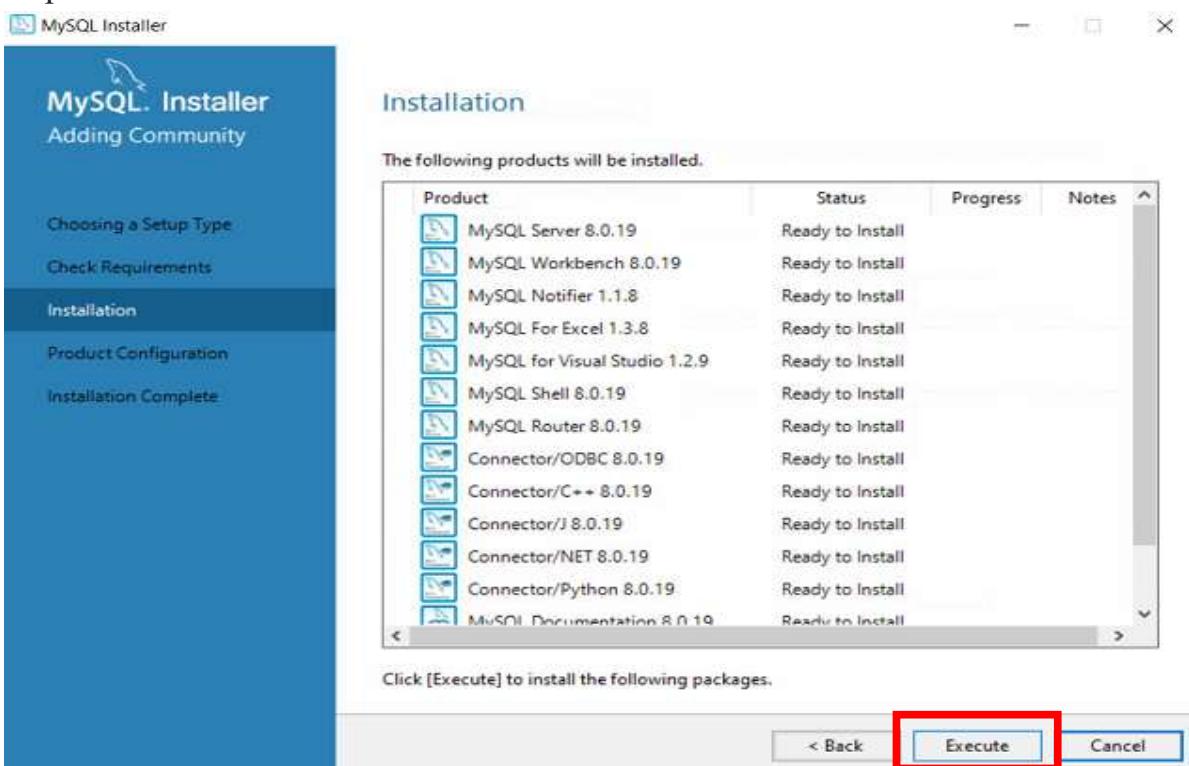
Step 7: Requirements also include Visual Studio 2015, 2017 or 2019. So if we do not have that, it should be installed from <https://visualstudio.microsoft.com/downloads/> community version.



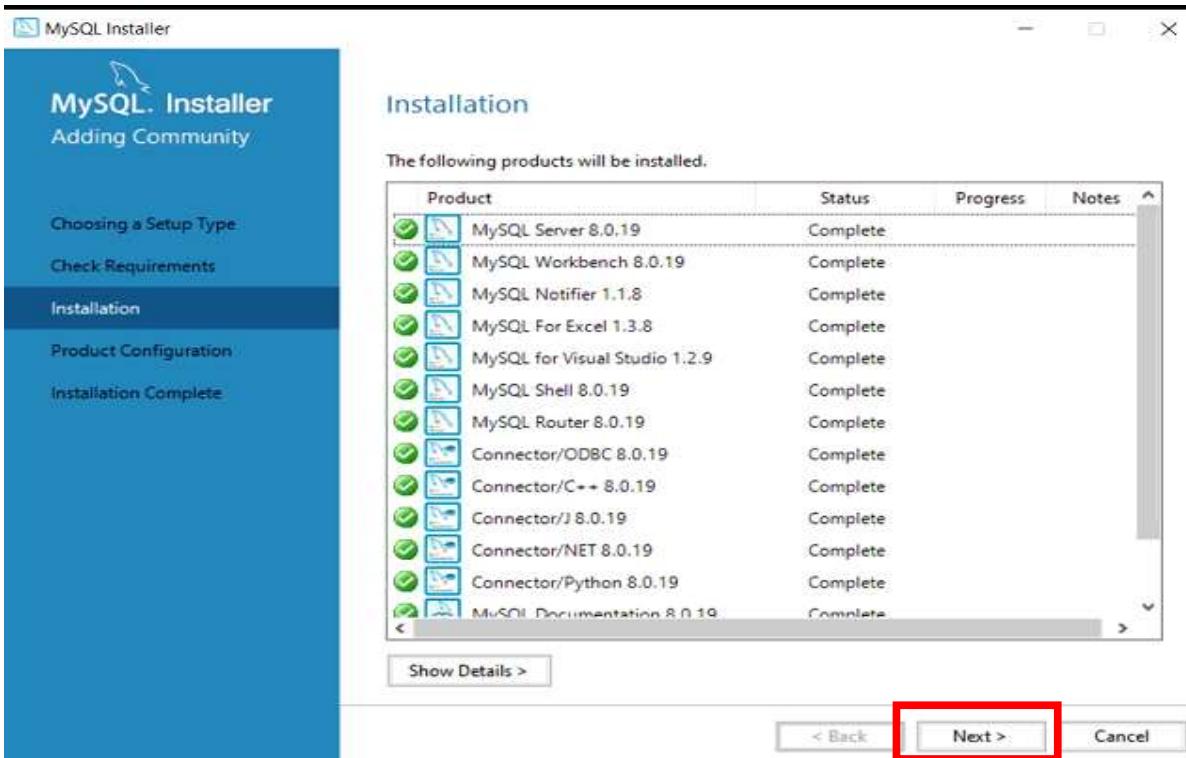
Step 8: After installing “Visual Studio 2015, 2017 or 2019” click on check. After all the requirements are satisfied click on next.



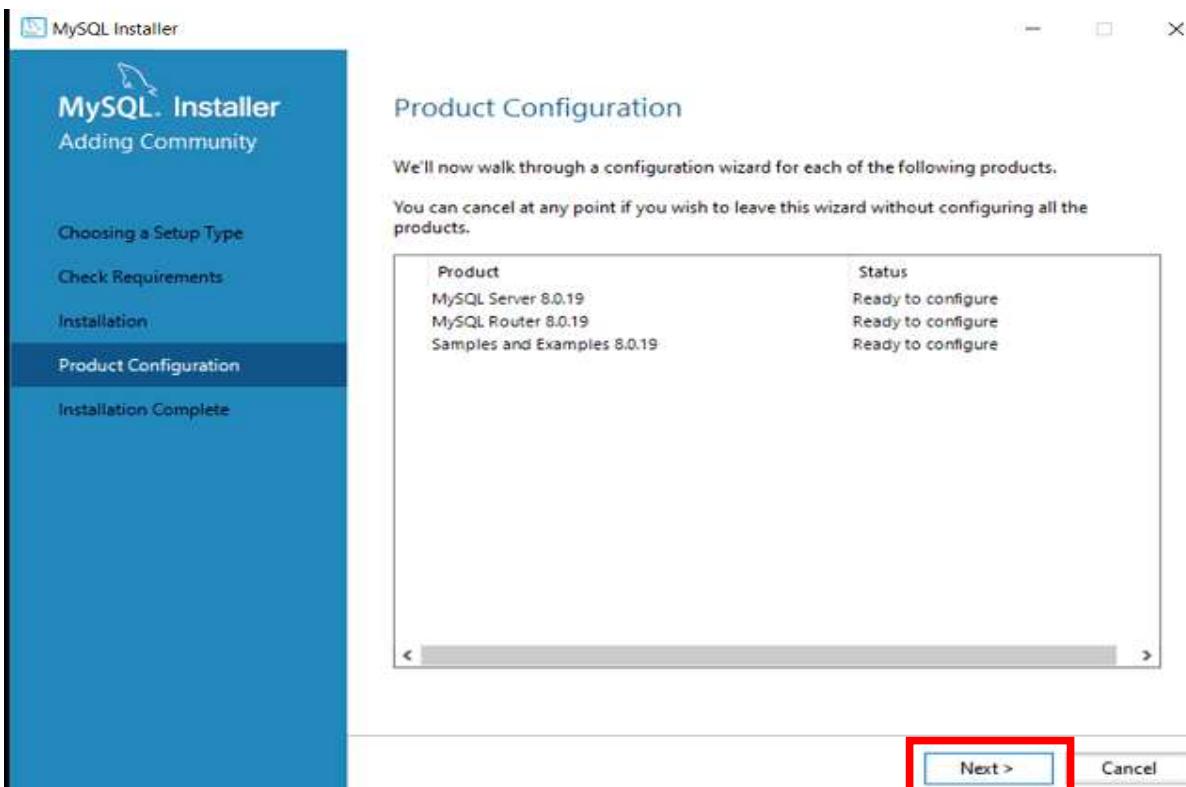
Step 9: Click on Execute.

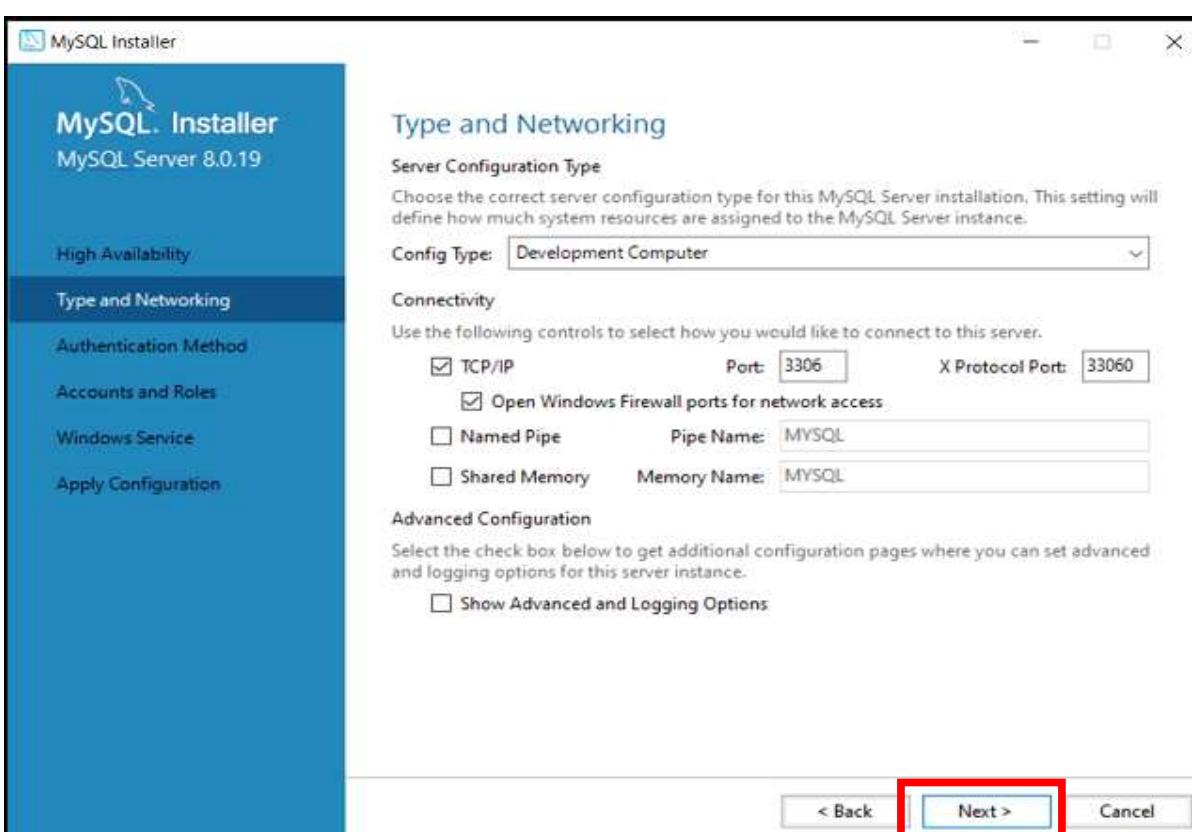
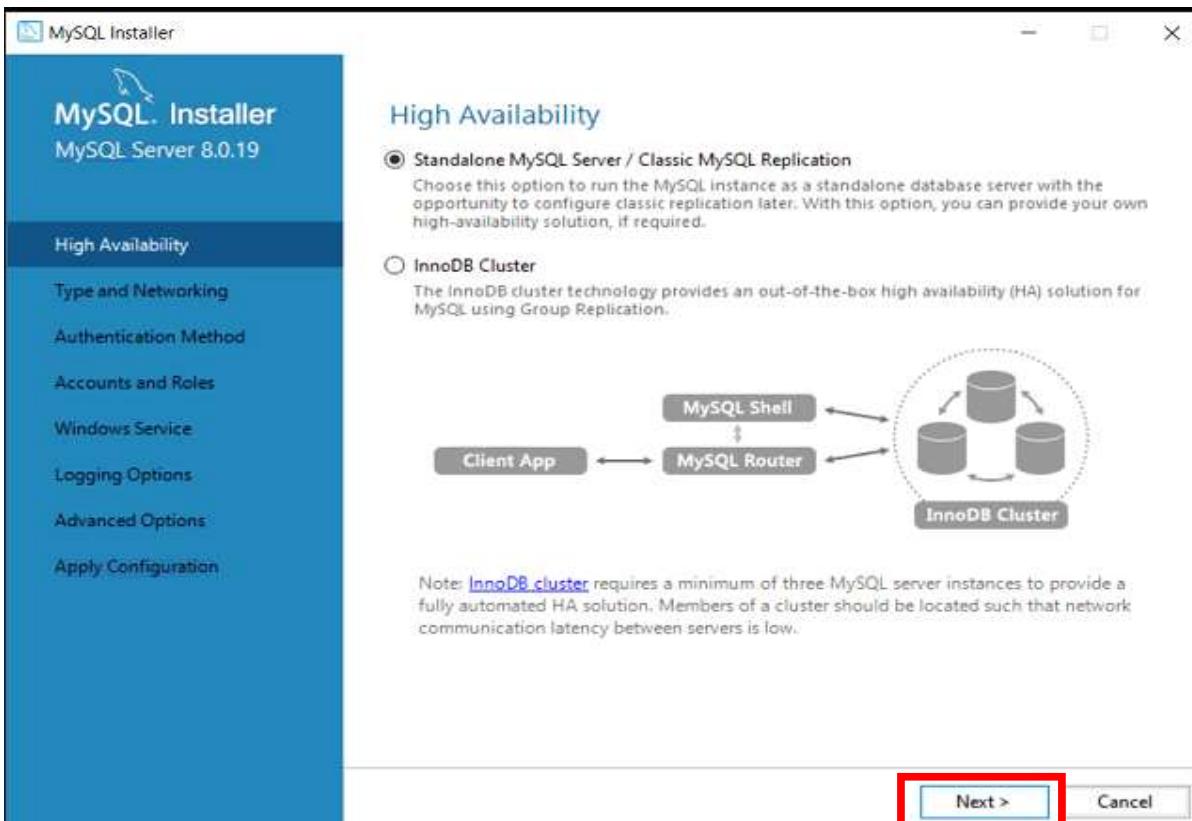


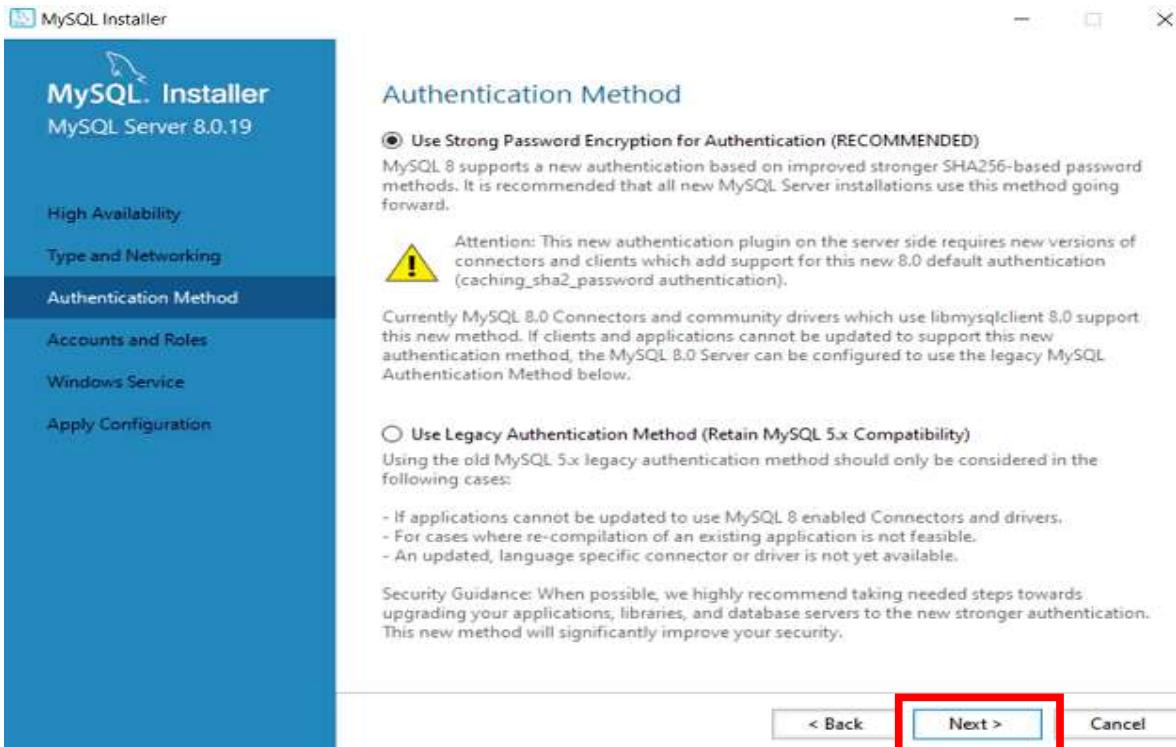
Step 10: After installing all the MySQL products, click on next.



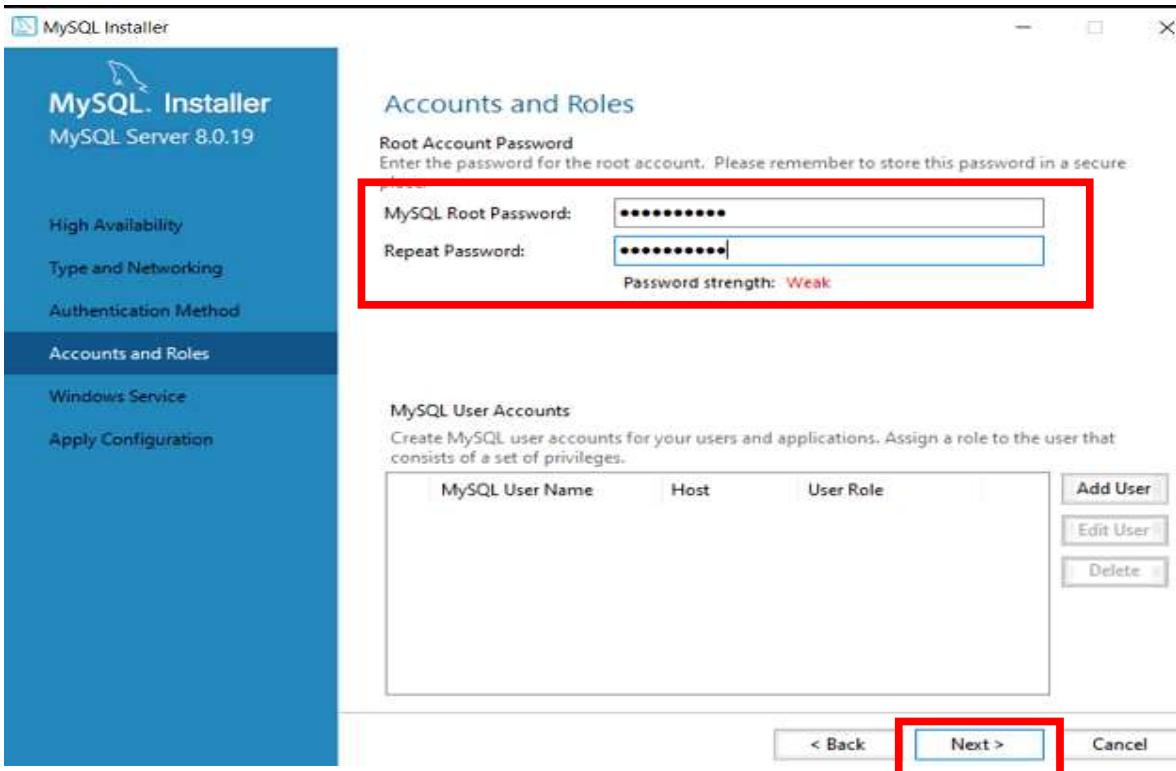
Step 11: Keep on clicking next for all the below-given steps.



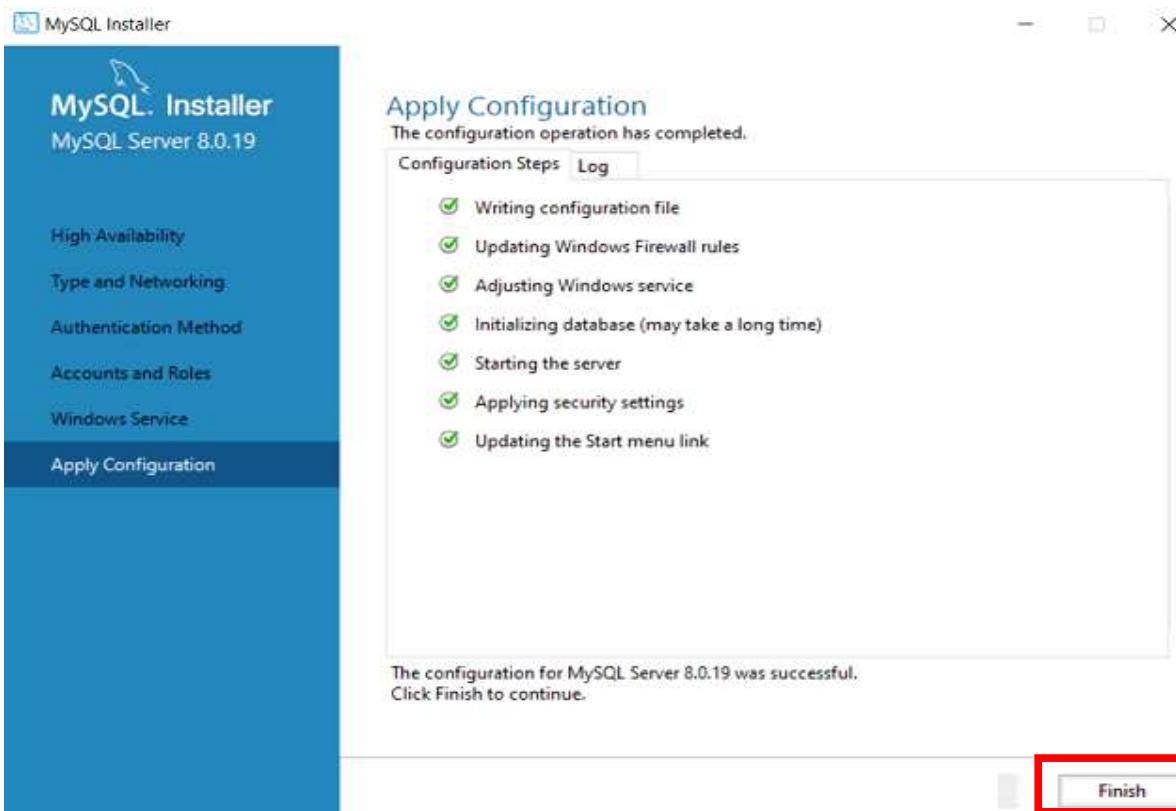
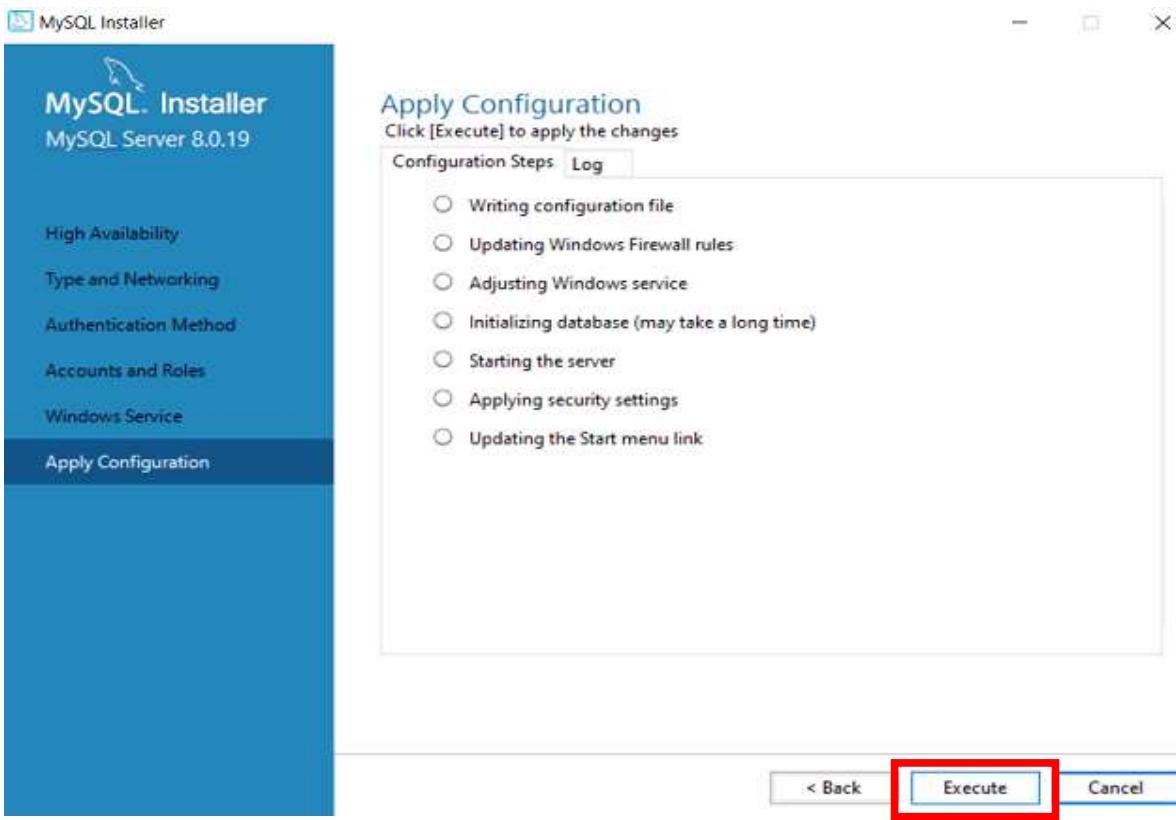




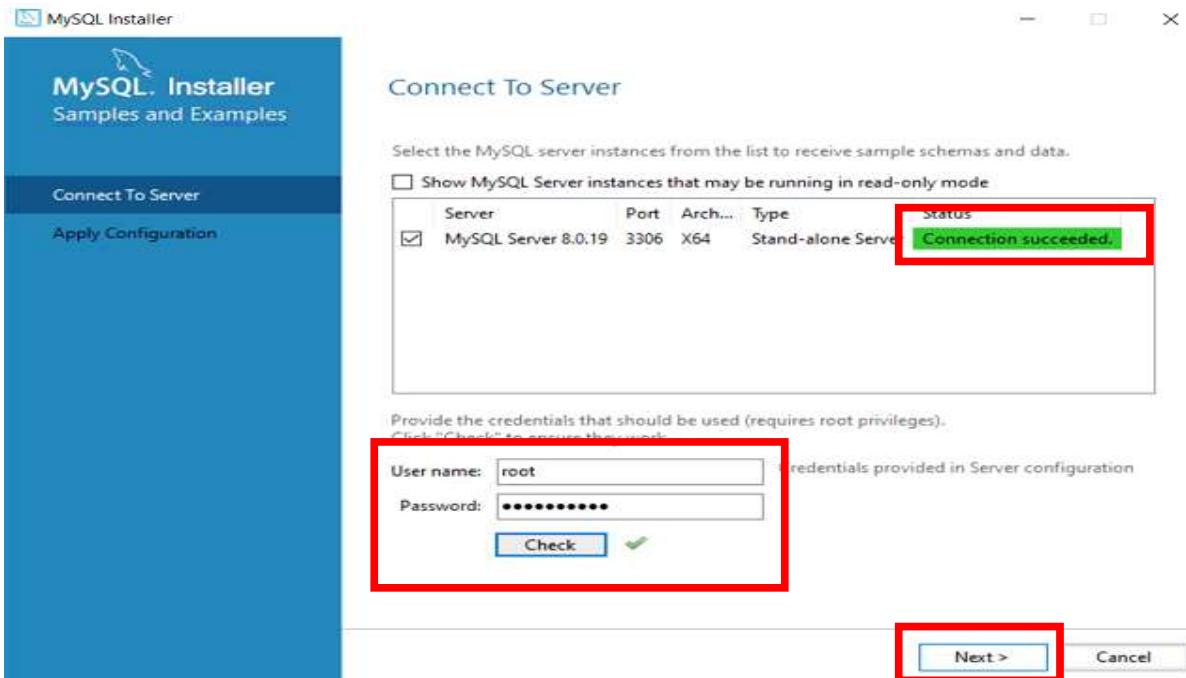
Step 12: Now set a strong password which you should remember and click on next.



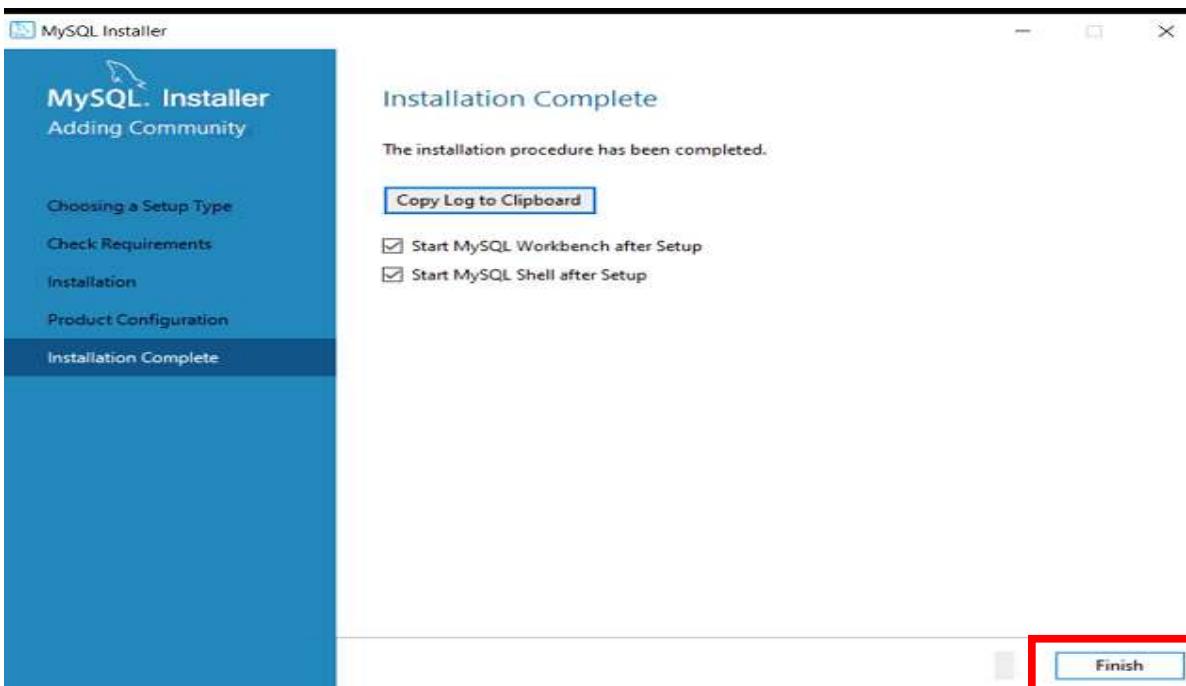
Step 13: Click on execute and then finish.



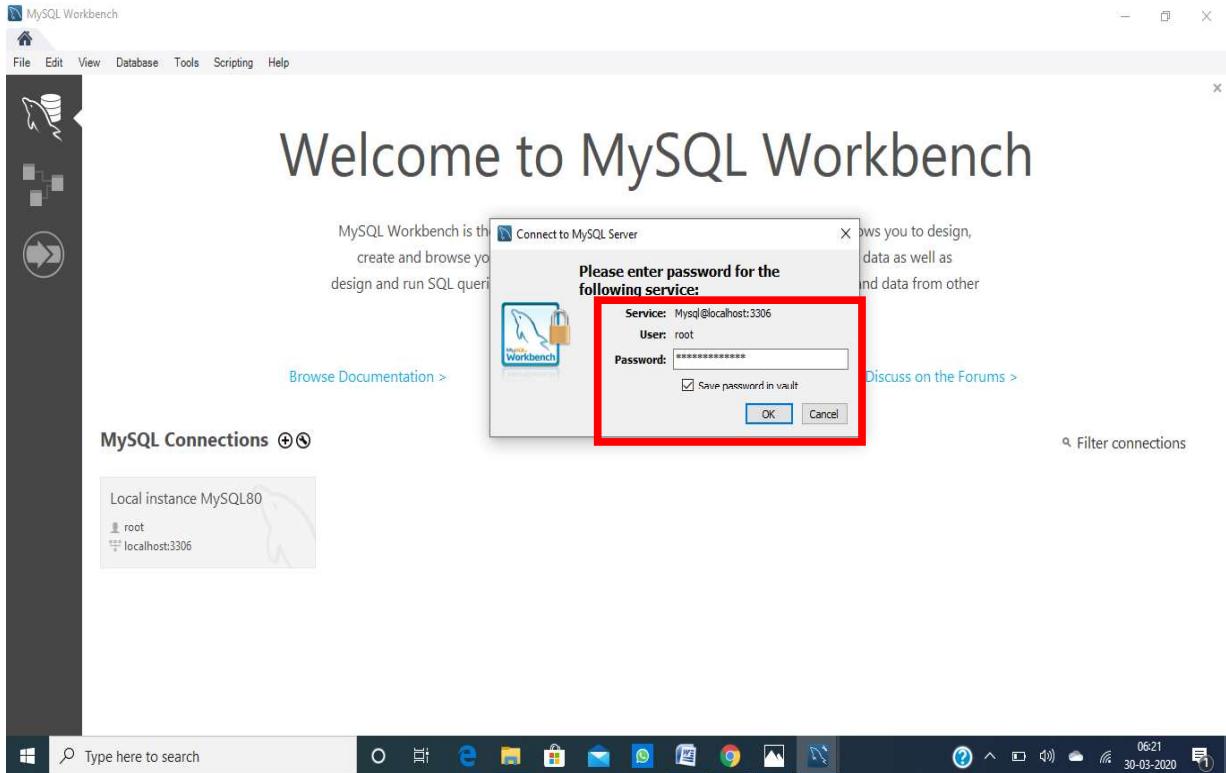
Step 14: For connecting to the MySQL server enter the password which you have created in one of the previous steps and click on check, if the password is correct then the connection to the MySQL server will be succeeded.



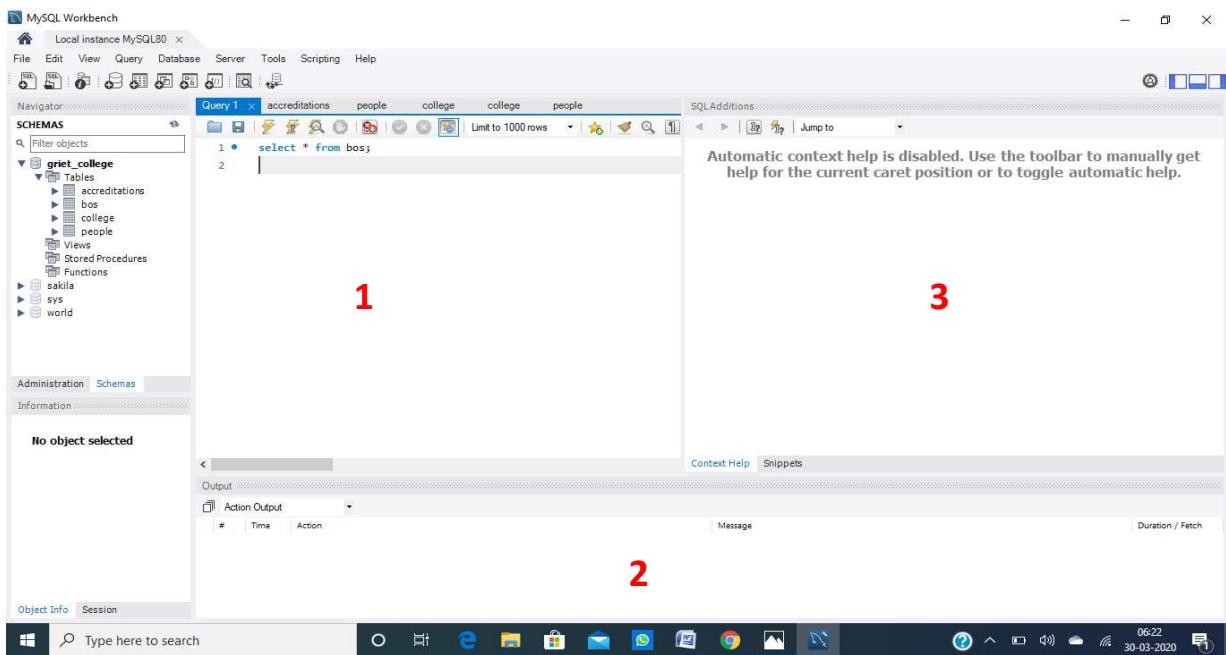
Step 15: Now, click on finish.



Step 16: After clicking on finish MySQL workbench will be opened. Now click on local instance and type the password which we have created. Select the check box for saving the password for future logins and click on ok.



Step 17: SQL editor will be opened for running SQL queries. It contains three panels.

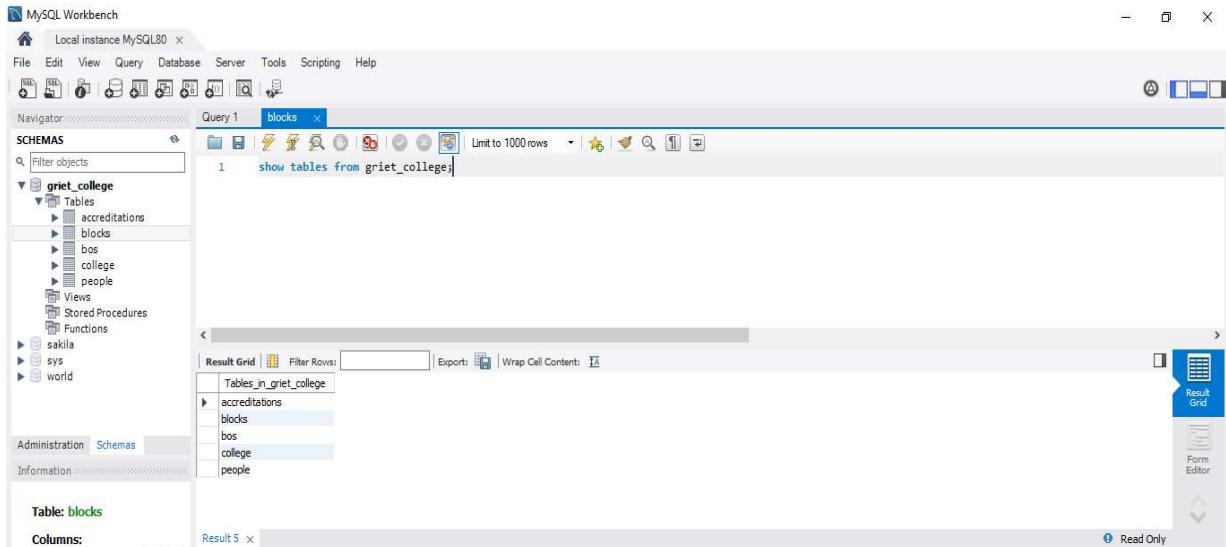


Database creation

(8) We need to create a database in the MySQL Workbench. Using the Structured Query Language we create the required tables. As of now the tables which we have created are shown in the below pictures.

Database → griet_college

Tables → accreditations, blocks, bos(Board of studies), college, people.



MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas Query 1 blocks

SCHEMAS Filter objects

griet_college Tables accreditations blocks bos college people

sakila sys world

Administration Schemas Information

Table: blocks

Columns:

Result 5 ×

Query 1 blocks

1 show tables from griet_college;

Result Grid Filter Rows: Export: Wrap Cell Content: Tables_in_griet_college

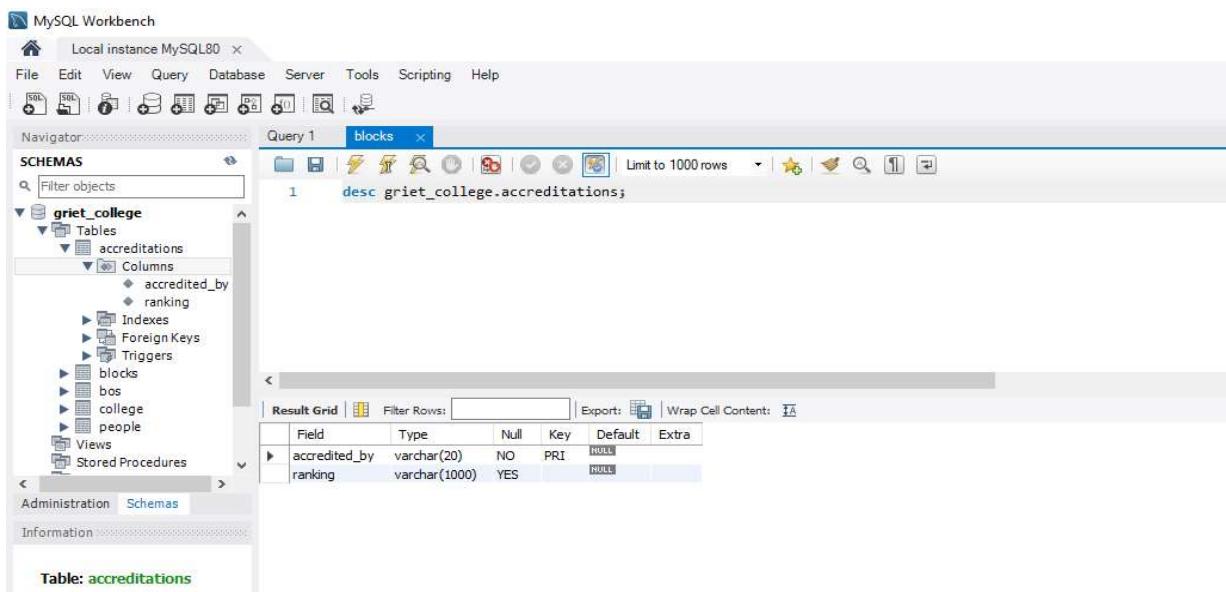
accreditations blocks bos college people

Result Grid Form Editor

Read Only

Table → accreditations

Columns → accredited_by, ranking



MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas Query 1 blocks

SCHEMAS Filter objects

griet_college Tables accreditations Columns accredited_by ranking

blocks bos college people

Views Stored Procedures

Administration Schemas Information

Table: accreditations

Query 1 blocks

1 desc griet_college.accreditations;

Result Grid Filter Rows: Export: Wrap Cell Content: Fields

Field	Type	Null	Key	Default	Extra
accredited_by	varchar(20)	NO	PRI	NULL	
ranking	varchar(1000)	YES		NULL	

Table→blocks

Columns→block_name, block_no, hod_room_no

The screenshot shows the MySQL Workbench interface with the 'blocks' table selected in the 'griet_college' schema. The 'Tables' section of the Navigator pane shows the 'blocks' table, which has three columns: 'block_name', 'block_no', and 'hod_room_no'. The 'Result Grid' pane displays the table structure with the following data:

Field	Type	Null	Key	Default	Extra
block_name	varchar(30)	NO	PRI	NULL	
block_no	int	NO		NULL	
hod_room_no	int	YES		NULL	

Table→bos

Columns→programme, chairperson_name, qualifications.

The screenshot shows the MySQL Workbench interface with the 'bos' table selected in the 'griet_college' schema. The 'Tables' section of the Navigator pane shows the 'bos' table, which has three columns: 'programme', 'chairperson_name', and 'qualifications'. The 'Result Grid' pane displays the table structure with the following data:

Field	Type	Null	Key	Default	Extra
programme	varchar(20)	NO	PRI	NULL	
chairperson_name	varchar(45)	NO		NULL	
qualifications	varchar(1000)	YES		NULL	

Table → college

Columns → about, info

The screenshot shows the MySQL Workbench interface with the 'griet_college' schema selected in the Navigator pane. The 'college' table is expanded, showing its columns: 'about' and 'info'. A query window titled 'Query 1' contains the command 'desc griet_college.college;'. The result grid displays the column details:

Field	Type	Null	Key	Default	Extra
about	varchar(20)	NO	PRI	NULL	
info	varchar(1000)	NO		NULL	

Table → people

Column → position, person_name, qualifications, about

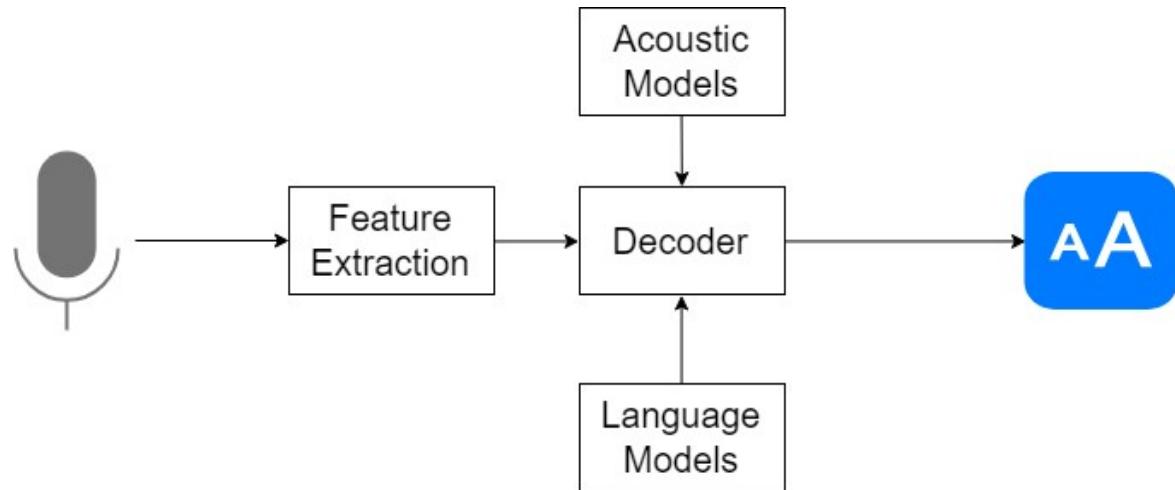
The screenshot shows the MySQL Workbench interface with the 'griet_college' schema selected in the Navigator pane. The 'people' table is expanded, showing its columns: 'position', 'person_name', 'qualifications', and 'about'. A query window titled 'Query 1' contains the command 'desc griet_college.people;'. The result grid displays the column details:

Field	Type	Null	Key	Default	Extra
position	varchar(20)	NO	PRI	NULL	
person_name	varchar(30)	NO		NULL	
qualifications	varchar(500)	YES		NULL	
about	varchar(2000)	YES		NULL	

2.5 Schematic Diagrams

2.5.1 Speech-to-Text conversion

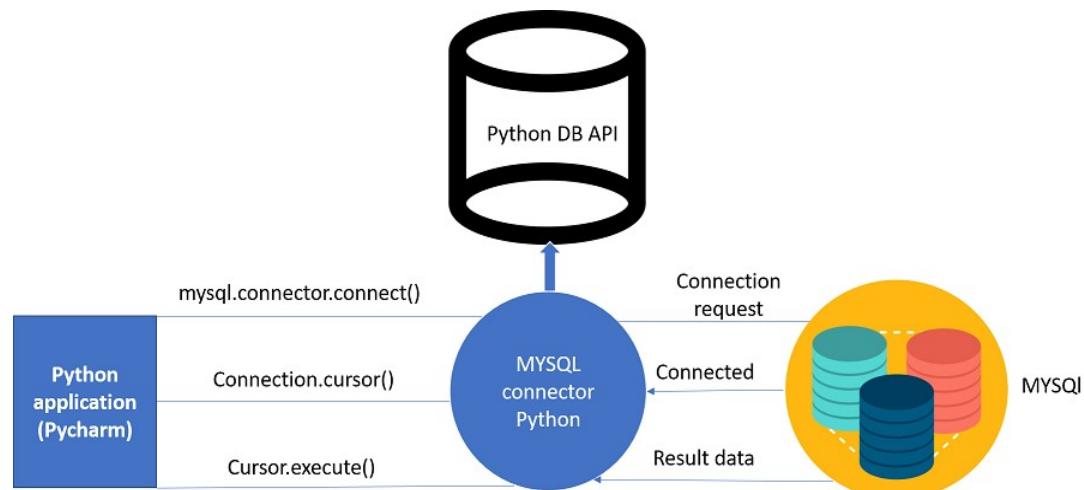
(⁹)Fig 5: Speech-to-text



Microphone is the source for input speech which undergoes certain processes like Feature Extraction, Acoustic models and language models which further gets decoded using a decoder and gets converted into text format.

2.5.2 Integration of Python and MySQL

(¹⁰)Fig 6: Integration of Python and MySQL

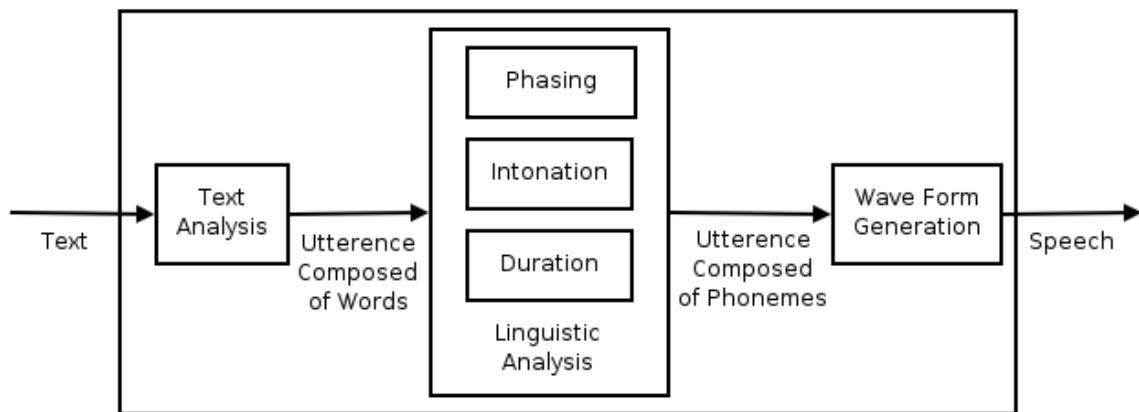


The integration of Python and MySQL will be done using the Python DB API. The Python application sends a connection request to the MySQL server. A cursor is

created after the connection is established successfully. Using this cursor we run SQL queries from the Python file.

2.5.2 Text-to-Speech conversion

⁽¹¹⁾Fig 7: Text-to-speech conversion

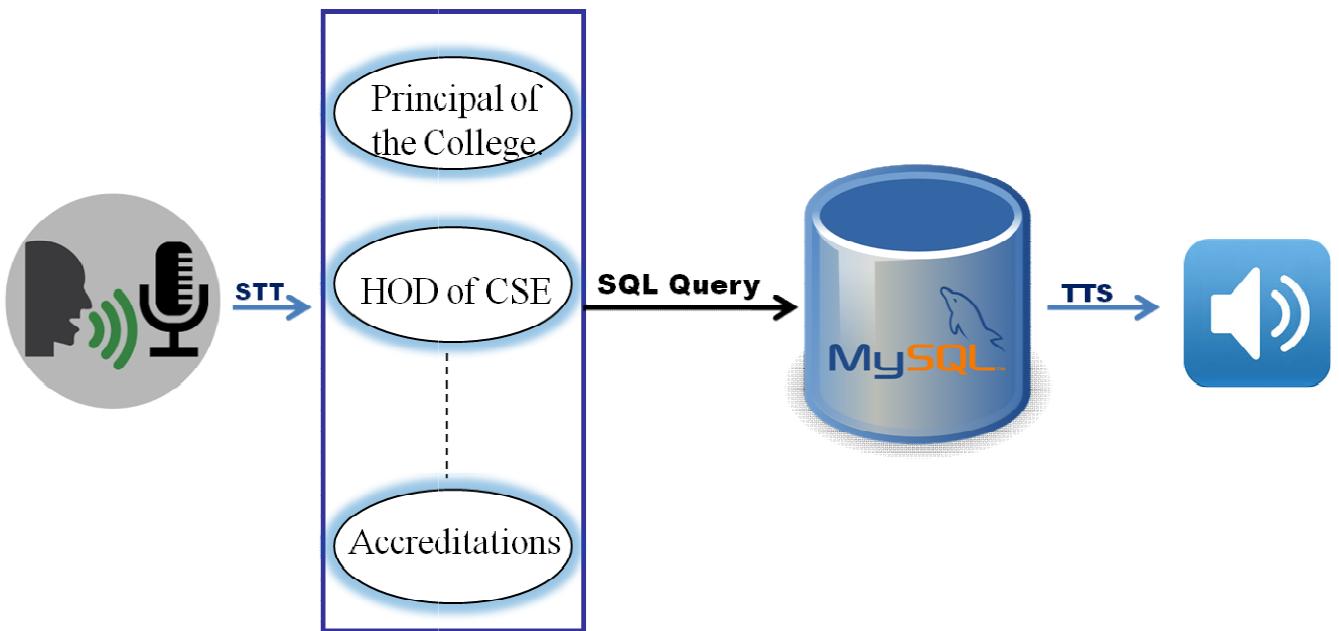


The text which had to be converted into speech will be analyzed and creates utterance composed of words. After the Linguistic analysis the utterance composed of phonemes will be created and then generates wave form. This wave form is the required speech format.

3. IMPLEMENTATION

3.1 Architecture

Fig 8: Architecture



The user gives a speech as input through the microphone; this speech is converted into text and the appropriate SQL query is applied to the created Database which contains details of the college. The response of the SQL query will be in text format, now this text is converted into speech and given as output through the speaker.

3.2 Modules description

There are three modules for implementing this project.

1. Speech-to-text conversion.
2. Integration of Python with Database.
3. Text-to-speech conversion.

3.2.1 Speech-to-text conversion

For converting the input that is speech into text, we use a Python library called SpeechRecognition. Speech recognition is the ability of computer software to identify words and phrases in spoken language and convert them into human- readable text. So we do not need to build any machine learning model from scratch, this library provides us with convenient wrappers for various well known public speech recognition APIs (such as Google Cloud Speech API, IBM Speech To Text, etc.).

The SpeechRecognition library should be installed in the PyCharm Terminal using pip. The command for windows is **>>pip install SpeechRecognition**.

Now, the input that is speech will be through the microphone. So to use the Microphone as the input source we need to install PyAudio which contains a microphone class.

The PyAudio Library also should be installed in the PyCharm terminal using the following command **>>pip install pyaudio**

We use the Microphone() object to read the audio from the default microphone, and then we used the timeout parameter in listen() function to stop reading after 8 seconds and then uploads the audio data to Google to get the output text.

Adjust_for_ambient_noise(source) is to listen for 1 sec calibrate the energy threshold for ambient noise levels.

We recognize different languages by passing language parameter to recognize_google() function. It also takes the audio as a parameter and returns a text which will be stored in a variable ‘sentence’.

3.2.2 Integration of Python and Database

To connect the Python application with the database we use MySQL-connector library which has to be installed in the PyCharm terminal using the following command **>>pip install MySQL-connector**

The MySQL.connector has a connect() method which takes host, user, passwd, database as its parameters where host is our localhost user is the current database user, passwd is the password for that particular user and the database created by that user for which the connection has to be established.

If the connection to MySQL server is successful then it prints “successfully connected to Database” or else it prints “Database connection Unsuccessful”

The cursor object of connector class is used to execute the SQL query.

3.2.3 Text-to-speech Conversion

The output of the SQL query will be a tuple. So this tuple should be converted into text and further this text is converted into an audio file. There are several APIs available to convert text to speech in Python. One such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as an mp3 file.

The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow.

The gTTS module is installed using the following command in the PyCharm terminal
`>>pip install gTTS`

The audio file created will be played using the Python playsound module. The gTTS contains three parameters. Text is the text which has to be converted to speech, lang is the language in which it has to be converted that is English. Slow, if it is true then the audio will play in slow mode, if it is false then it will play in normal mode. The save() instance saves the audio file in mp3 format. And this saved audio file will be played using the playsound instance.

3.3 Input and output analysis

Seldom visitor of the college just have to ask the assistant (their required query) in speech format. The assistant listens and responds with an appropriate answer for the asked query in speech format itself.

4. CONCLUSION AND FUTURE ENHANCEMENT

4.1 Conclusion

In conclusion, “GRIET Assistant” will respond only to limited queries because the database doesn’t contain all the information about the college. It has the following information.

1. NBA, NIRF, NAAC, UGC accreditations.
2. Department block name, block number and HOD (Head of the Department) room number.
3. Board of studies; contains the name of the chairperson of a department and their qualifications. Departments include ‘Basic Sciences and Humanities’, ‘Chemistry’, ‘Civil’, ‘Computer Science’, Electronics and communication engineering’, ‘Electrical and electronics engineering’, ‘English’, ‘Humanities’.
4. Data in the table college include Core Values, History, Mission, Quality Policies, Special days, Strategies, Student Events, and Vision of the college.
5. The table named people consists of name, qualifications and about the info of Chairman, Director, Principal and Vice president of the college.

4.2 Future Enhancement

The actual idea behind implementing this kind of project is Amazon Alexa and Google Assistant. Amazon Alexa is used in Amazon echo smart speaker, which just looks like a speaker. When we operate it, we feel like speaking to a normal person rather than a speaker. It makes our work simple and fast without any effort.

So this “GRIET Assistant” can also be embedded into a hardware device that is made with the help of Arduino containing a microphone as a source of input and a speaker as the source of output.

The data in the database can be extended which includes each every detail about the college. As of now the Database contains the data which is fixed that is which will be unchanged for a long time, but for better usage or to make the application more useful we can also include the inconsistent data. The addition of inconsistent data brings a change in the coding and implementation part.

Bibliography

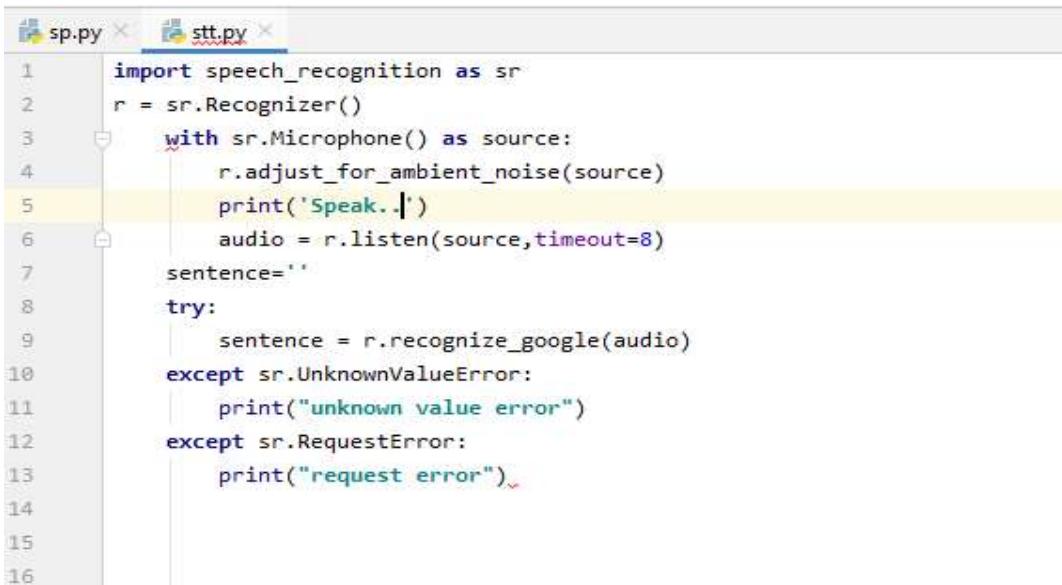
1. Data Science. *Encyclopedia*. [Online] April 17, 2020.
https://en.wikipedia.org/wiki/Data_science.
2. **Yse, Diego Lopez.** Your Guide to Natural Language Processing (NLP). *Medium*. [Online] January 16, 2019. <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>.
3. Installing Python. [Online] <https://www.guru99.com/how-to-install-python.html>.
4. PyCharm wiki. *Encyclopedia*. [Online] April 14, 2020.
<https://en.wikipedia.org/wiki/PyCharm>.
5. **Guru99.** Installing PyCharm. [Online] <https://www.guru99.com/how-to-install-python.html>.
6. MySQL Workbench wiki. *Encyclopedia*. [Online] December 17, 2019.
https://en.wikipedia.org/wiki/MySQL_Workbench.
7. Dev MYSQL. *MySQL Community Downloads*. [Online]
<https://dev.mysql.com/downloads/workbench/>.
8. Database creation in MySQL Workbench. *TechGeekArti*. [Online] December 28, 2018. <https://www.youtube.com/watch?v=pZKrznZymMg&t=585s>.
9. **SABA, Gulsanober.** Speech Recognition. *Simplified Python*. [Online] 2017.
<https://www.simplifiedpython.net/speech-recognition-python/>.
10. **kant, Harshit.** Python Database Connection. *edureka*. [Online] 2019.
<https://www.edureka.co/blog/python-database-connection/>.
11. Python Text to Speech. *deparkes*. [Online] 2017.
<https://deparkes.co.uk/2017/06/30/python-text-speech/>.

APPENDICES

Plagiarism Report:

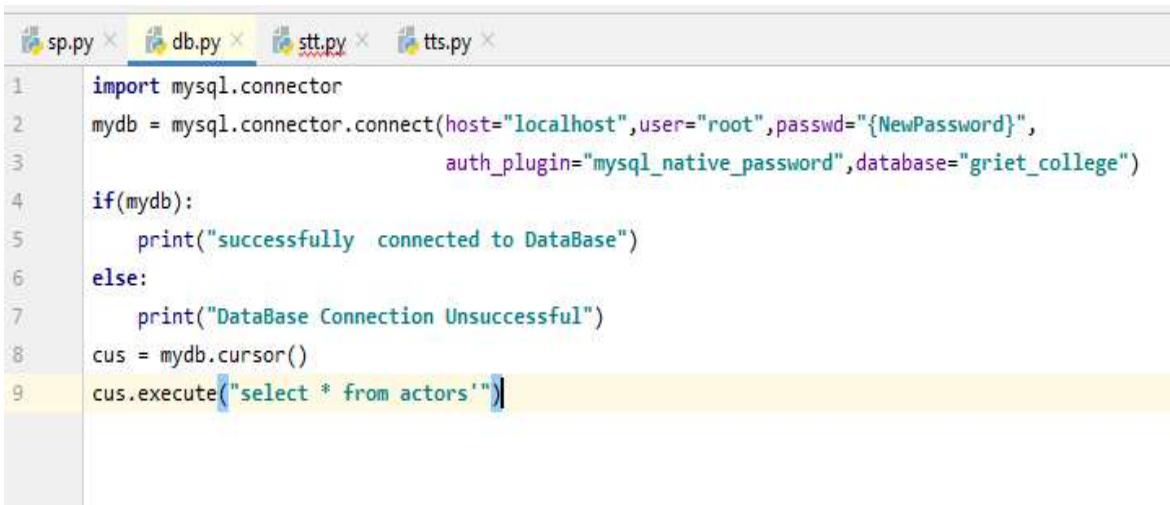
Screen shots:

1. Speech-to-text



```
sp.py x stt.py x
1 import speech_recognition as sr
2 r = sr.Recognizer()
3 with sr.Microphone() as source:
4     r.adjust_for_ambient_noise(source)
5     print('Speak.')
6     audio = r.listen(source,timeout=8)
7 sentence=''
8 try:
9     sentence = r.recognize_google(audio)
10 except sr.UnknownValueError:
11     print("unknown value error")
12 except sr.RequestError:
13     print("request error")
14
15
16
```

2. Connecting to MySQL server



```
sp.py x db.py x stt.py x tts.py x
1 import mysql.connector
2 mydb = mysql.connector.connect(host="localhost",user="root",passwd="{NewPassword}",
3                                 auth_plugin="mysql_native_password",database="griet_college")
4 if(mydb):
5     print("successfully connected to DataBase")
6 else:
7     print("DataBase Connection Unsuccessful")
8 cus = mydb.cursor()
9 cus.execute("select * from actors")
```

3. Text-to-speech

The screenshot shows a PyCharm interface with four tabs at the top: sp.py, db.py, stt.py, and tts.py. The tts.py tab is active, displaying the following Python code:

```
1 from gtts import gTTS
2 import playsound
3
4 myText="Social Distancing is the best to "
5 language='en'
6
7 output=gTTS(text=myText,lang=language,slow=False)
8
9 output.save("output.mp3")
10
11 playsound.playsound("output.mp3")
12
```

Output:

The screenshot shows the PyCharm interface with the sp.py tab active in the project navigation bar. The code editor displays the sp.py script, which contains a function recordAudio(). The run output window below shows the execution of the script and its response:

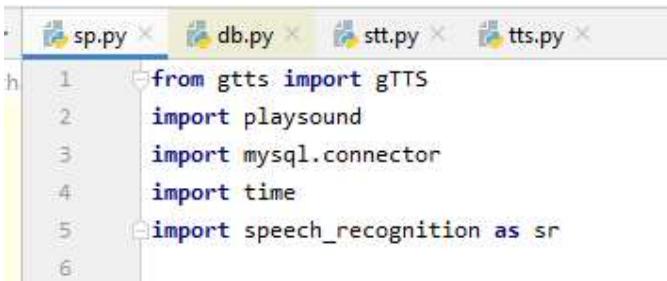
```
C:\Users\Shravani\PycharmProjects\untitled\venv\Scripts\python.exe C:/Users/Shravani/PycharmProjects/untitled/sp.py
wake up the assistant and then ask your question
wakeup words: 'hello tech', 'hello champion', 'hello assistant'
Speak..
You said: hello tech who is the principal of the college
here is your response..
Dr. J Praveen

Process finished with exit code 0
```

The status bar at the bottom indicates PEP 8: expected 2 blank lines after class or function definition, found 1.

Sample codes:

Now coming to Python code, the libraries imported are as follows

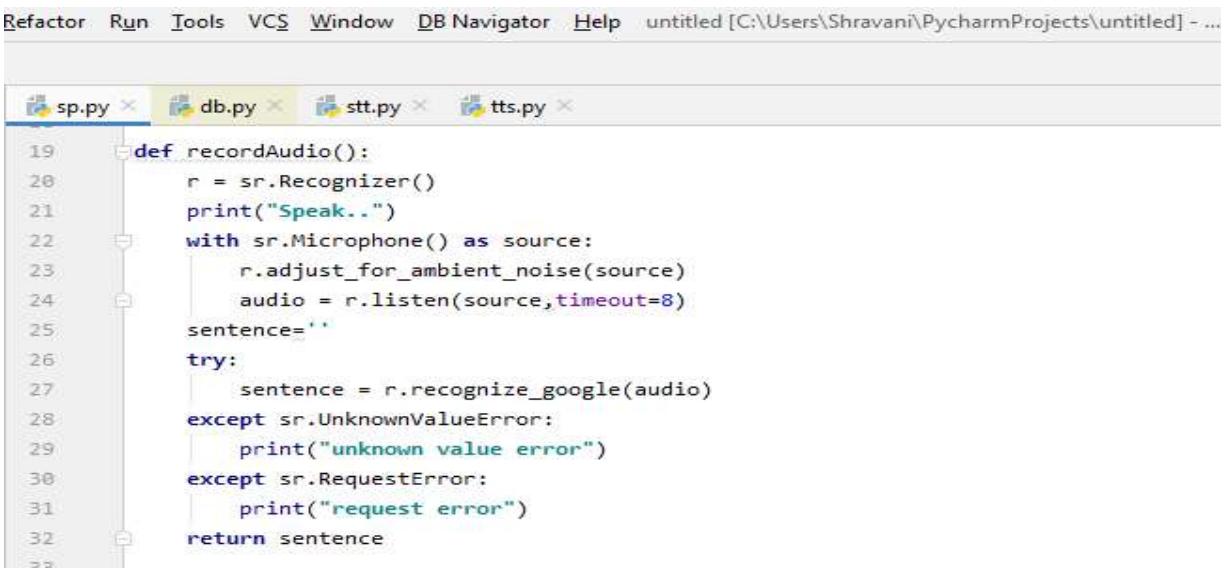


```
1 from gtts import gTTS
2 import playsound
3 import mysql.connector
4 import time
5 import speech_recognition as sr
```

Next we have the code to get connected with the Database.

```
6 mydb = mysql.connector.connect(host="localhost",user="root",passwd={"NewPassword"},
7                                 auth_plugin="mysql_native_password",database="griet_college")
8
9
10 if(mydb):
11     #print("successfully connected to DataBase")
12     print('wake up the assistant and then ask your question')
13     print("wakeup words: 'hello tech','hello champion','hello assistant'")
14
15 else:
16     print("DataBase Connection Unsuccessful")
17 cus = mydb.cursor()
18
```

The recordAudio() function takes the input which is in speech format through the microphone and converts that audio format into text format and returns this converted text.

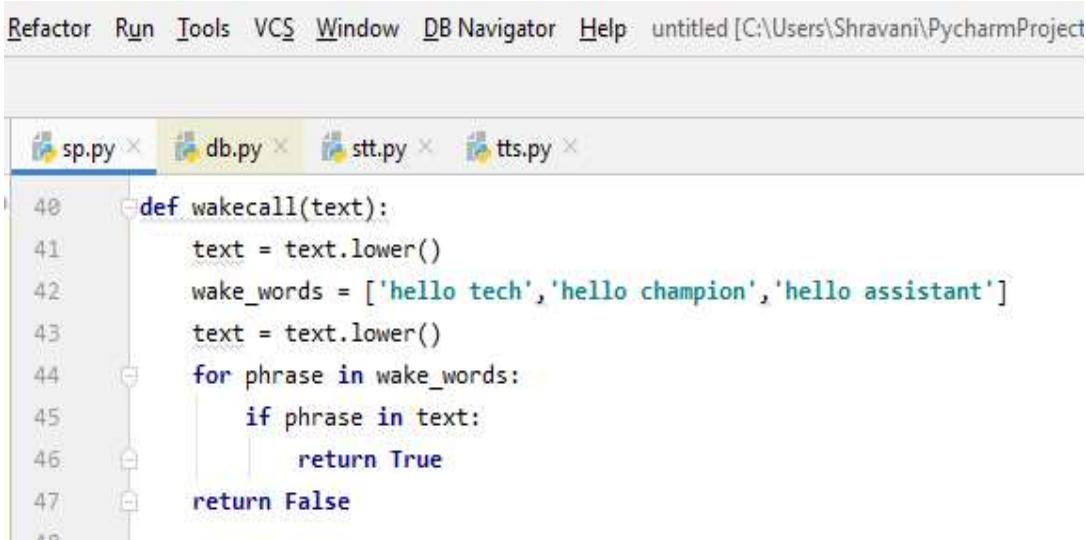


```
19 def recordAudio():
20     r = sr.Recognizer()
21     print("Speak..")
22     with sr.Microphone() as source:
23         r.adjust_for_ambient_noise(source)
24         audio = r.listen(source,timeout=8)
25         sentence=''
26     try:
27         sentence = r.recognize_google(audio)
28     except sr.UnknownValueError:
29         print("unknown value error")
30     except sr.RequestError:
31         print("request error")
32     return sentence
```

The response() takes the text as a parameter, converts it into an audio file and plays this converted audio file. The parameter for this function will be the response of the database query.

```
33
34 def response(mytext):
35     language = 'en'
36     output = gTTS(text=mytext, lang=language, slow=False)
37     output.save("output.mp3")
38     playsound.playsound("output.mp3")
39
```

The wakecall() has a parameter text. The input audio is converted to text and sent as a parameter for this function. It checks whether the user has given the valid input wakeup word or not.



```
Refactor Run Tools VCS Window DB Navigator Help untitled [C:\Users\Shravani\PycharmProject]
sp.py × db.py × stt.py × tts.py ×
40 def wakecall(text):
41     text = text.lower()
42     wake_words = ['hello tech', 'hello champion', 'hello assistant']
43     text = text.lower()
44     for phrase in wake_words:
45         if phrase in text:
46             return True
47     return False
```

Next we have the Driver code followed by if statements in which we check the wakeup word and then run the required SQL query.

```

48     flag=1
49     time.sleep(3)
50     text=recordAudio()
51     text = text.lower()
52     print("You said: ", text)
53
54     if wakecall(text):
55
56         if 'history' in text and ('college' in text or 'griet' in text):
57             cus.execute("select info from college where about='History'")
58
59         elif 'vision' in text and ('college' in text or 'griet' in text):
60             cus.execute("select info from college where about='Vision'")
61
62         elif 'mission' in text and ('college' in text or 'griet' in text):
63             cus.execute("select info from college where about='Mission'")
64
65         elif ('quality policies' in text or 'quality policy' in text) and ('college' in text or 'griet' in text):
66             cus.execute("select info from college where about='QualityPolicy'")
67
68         elif ('strategies' in text or 'strategy' in text) and ('college' in text or 'griet' in text):
69             cus.execute("select info from college where about='Strategies'")
70
71         elif 'core values' in text and ('college' in text or 'griet' in text):
72             cus.execute("select info from college where about='CoreValues'")
73
74         elif 'student events' in text and ('college' in text or 'griet' in text):
75
76             if wakecall(text)

```

The SQL query returns a tuple. This tuple is converted to String and sent as a parameter for response() method. This method converts this text into audio file mostly mp3 format and plays it using the default audio player.

```

213         cus.execute("select hod_room_no from blocks where block_name='eee' ")
214     elif 'block' in text and ('mechanical' in text) and ('head' in text or 'hod' in text):
215         cus.execute("select hod_room_no from blocks where block_name='mech' ")
216     elif 'block' in text and ('it' in text or 'information technology' in text) and ('head' in text or 'hod' in text):
217         cus.execute("select hod_room_no from blocks where block_name='it' ")
218     elif 'block' in text and ('humanities' in text) and ('head' in text or 'hod' in text):
219         cus.execute("select hod_room_no from blocks where block_name='hs' ")
220     else:
221         flag=0
222
223     else:
224         print("invalid wakeup call..so repeat it again")
225         exit()
226
227     mytext=""
228     for db in cus:
229         mytext=''.join(db)
230     if flag==1:
231         print("here is your response..")
232         print(mytext)
233     else:
234         response(mytext)
235
236

```

