Industrial Training Report
On

Python programming

(2022-2023)

Submitted by

Shivansh Verma

Submitted to the

Department Of

Computer Science and Technology

SR INSTITUTE OF MANAGEMENT AND TECHNOLOGY

LUCKNOW

Affiliated to Dr. APJ ABDUL KALAM TECHNICAL UNIVERSITY (Lucknow)

CERTIFICATION

Science and Engineering successfully completed the training of six weeks on Programming with Python. At This is to certify that the Shivansh Verma , 1904850100109 of Computer Internshala.

INTERNSHALA TRAININGS

Certificate of Training

Shivansh Verma

from S.R. Institute Of Technology & Management has successfully completed a 6-week online training on **Programming with Python**. The training consisted of Introduction to Python, Using Variables in Python, Basics of Programming in Python, Principles of Object-oriented Programming (OOP), Connecting to SQLite Database, Developing a GUI with PyQT, Application of Python in Various Disciplines, and The Final Project modules. Shivansh scored 100% marks in the final assessment and is a top performer in the training. We wish Shivansh all the best for future endeavours.

Sarvesh Agarwal
FOUNDER & CEO, INTERNSHALA

Date of certification: 2022-09-12   Certificate no. : 4D3CD0B3-75EA-0D0F-EE92-3E3BD7757EE7

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

ACKNOWLEDGEMENT

It is my proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

First and foremost, I wish to record our sincere gratitude to Internshala Coordinators for their constant support and encouragement in preparation of this report and for making available videos and interface facilities needed to prepare this report.

The seminar on "Python" was very helpful to us in giving the necessary background information and inspiration in choosing this topic for the seminar. Their contributions and technical support in preparing this report are greatly acknowledged.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

Sincerely,

Shivansh verma

Index

1. Company Profile:

Introduction

Internshala is an online recruitment and training platform. On the recruitment portal of Internshala, the internship seekers and job-seeking freshers from all over India, across different education streams, can search and apply to various internships and fresher jobs of their choice and organizations. Additionally, start-ups, corporates, SMEs, NGOs, education institutes, and big brands, can post their intern and entry-level job requirements to hire university students and fresh graduates all over India On Internshala Trainings, the e-learning arm of Internshala, online learners including students and professionals can avail of online training in the latest in-demand industry skills to build their careers, upskill themselves, and become industry-ready.

Some Client Of The Company

Sarvesh is the one who started it all. It takes a lot of guts to create a start- and he's the one with all of it. Meet the CEO. Yes, he's got loads of patience and humility. But on that once-in-a-blue-moon day when he breathes fire, no one dares to tread his territory. He's a master manager and rightfully at the helm of affairs at Internshala.

Gaurav Sahu is the principal product manager for the e-learning product at Internshala. This month, we got in touch with him to learn about his younger self and his journey to becoming a principal product manager. Here's his story and advice for aspiring product managers out there.Business development manager

Sales Executive

Graphics designer clients

Social media managers

Interns clients

And also there are so many clients as a trainer who trained the student according to their skills and knowledge.

All these clients are very trained in their skills and knowledge. They enhance the skills of students.

Company Service in Technology

It is an education platform that provides online training .and it also provides service in IIT as a held workshop. Customer service skills are traits and practices that equip you to address customer needs and foster a positive experience. In general, customer service technical skills and improving coding.

This service provides is both a type of job and a set of job skills. As a job, customer service professionals are responsible for addressing customer needs and ensuring they have a good experience. As a skill set, customer service entails several qualities like active listening, empathy, problem-solving, and communication. All service is used in many jobs at every level.

Live Online Summer Training / Winter Camps / Project Based Trainings/ Workshops under International Certifications (MTA/EC Council/Adobe etc.)

2 days Technical Official Microsoft Technology Associate (MTA) Certified Workshop at All India IITs, NITs, and more renowned Colleges. Summer
Training / Winter Camps / Project Based Training under International Certifications (MTA/EC Council/Adobe etc
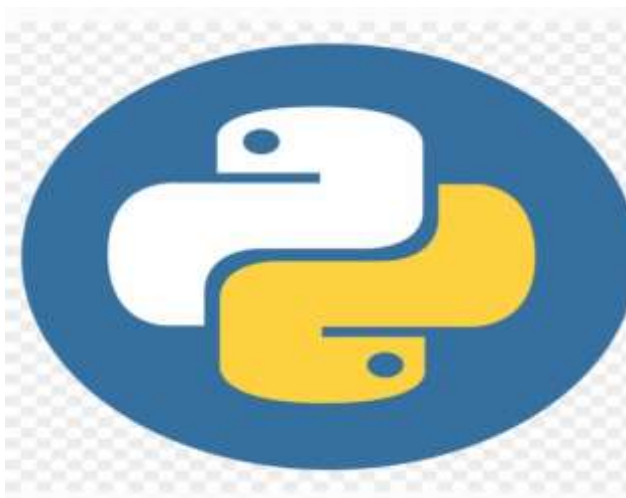
2. Technology

1. **Introduction to Python**

   Learn how to install Python, distinguish between important data types and use basic features of the Python interpreter, IDLE.

2. **Using Variables in Python**

   Learn about numeric, string, sequence and dictionary data types and relevant operations while practicing Python syntax.

3. **Basics of Programming in Python**

   Learn how to write programs using conditionals, loops, iterators and generators, functions and modules and packages.

4. **Principles of Object-oriented Programming (OOP)**

   Learn about the important features of Object-oriented Programming while using Classes and Objects, two main aspects of the OOP paradigm.

5. **Connecting to SQLite Database**

   Learn about relational databases while learning how to store and retrieve data from an SQLite database through Python.

6. **Developing a GUI with PyQT**

   Learn how to install PyQt5 toolkit, Qt Designer and create a graphical user interface using common widgets and menu systems.

7. **Application of Python in Various Disciplines**

   Learn about various resources to extend your learning for the Python programming language

   Python features include –

   • **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

   • **Easy-to-read** − Python code is more clearly defined and visible to the eyes.

   •**Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

   • **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

• **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
 • **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Background

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

Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

• **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program   before executing it. This is similar to PERL and PHP.

• **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

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

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

History of Python

 Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

 Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Objective

Learning Objectives

The learning objectives of this course are:

- To understand why Python is a useful scripting language for developers.

- To learn how to design and program Python applications.

- To learn how to use lists, tuples, and dictionaries in Python programs.

- To learn how to identify Python object types.

- To learn how to use indexing and slicing to access data in Python programs

- To define the structure and components of a Python program.

- To learn how to write loops and decision statements in Python.

- To learn how to write functions and pass arguments in Python.

- To learn how to build and package Python modules for reusability.

- To learn how to read and write files in Python.

- To learn how to design object-oriented programs with Python classes.

- To learn how to use class inheritance in Python for reusability.

Purpose and scope

Python can be an excellent, lucrative career, if you are passionate about developing websites, web apps, or doing UI/UX work. However, you need to frequently upgrade your skills as it is a continuously evolving field. Webandcrafts is one of the leading website development companies in Kerala, India.

Python with 1-4 years of experience can make around Rs 3,04,000 per annum. With 5-9 years of experience, you can earn around Rs 5,89,000 per annum. If you have an experience of over 10 years in the field, you can earn around Rs 1,000,000 per annum or even more depending on different factors. **Python allows you to express yourself creatively on the internet**. If you have an idea you would like to try as you discover your python skills, try it out. pyhton is a fun, creative experience.

There are numerous python jobs available in the market for those with the requisite skills. We discussed the trending technologies earlier –coders need to master those technologies: IoT, AI, VR, and AR, in addition to the latest programming languages, platforms, tools, and so on. If you aspire to be a python development professional, make sure you learn all these skills.

For any business, a website matters more than anything, when it comes to reaching out to customers online. A website is the representation of the business online. These days, every business realizes the need for having a website and are putting in efforts to design and develop the best site for taking their products or services online. This is where we can see a great deal of scope for web development and design. With the tremendous progress in the launch of websites, people who can create exemplary designs and platforms for their online presence are what businesses are searching for. python developers and designers are bringing in all the best of their technical expertise and skills to develop unique design websites that are capable of pulling the crowd.

Most web developers and designers are familiar with the latest tools, techniques, technologies, and frameworks that will help them get beautiful sites up in a short period. The development time of websites has come down with the agile technologies that make development easier. Here, we will cover the scope and future of web development.

Output Screen
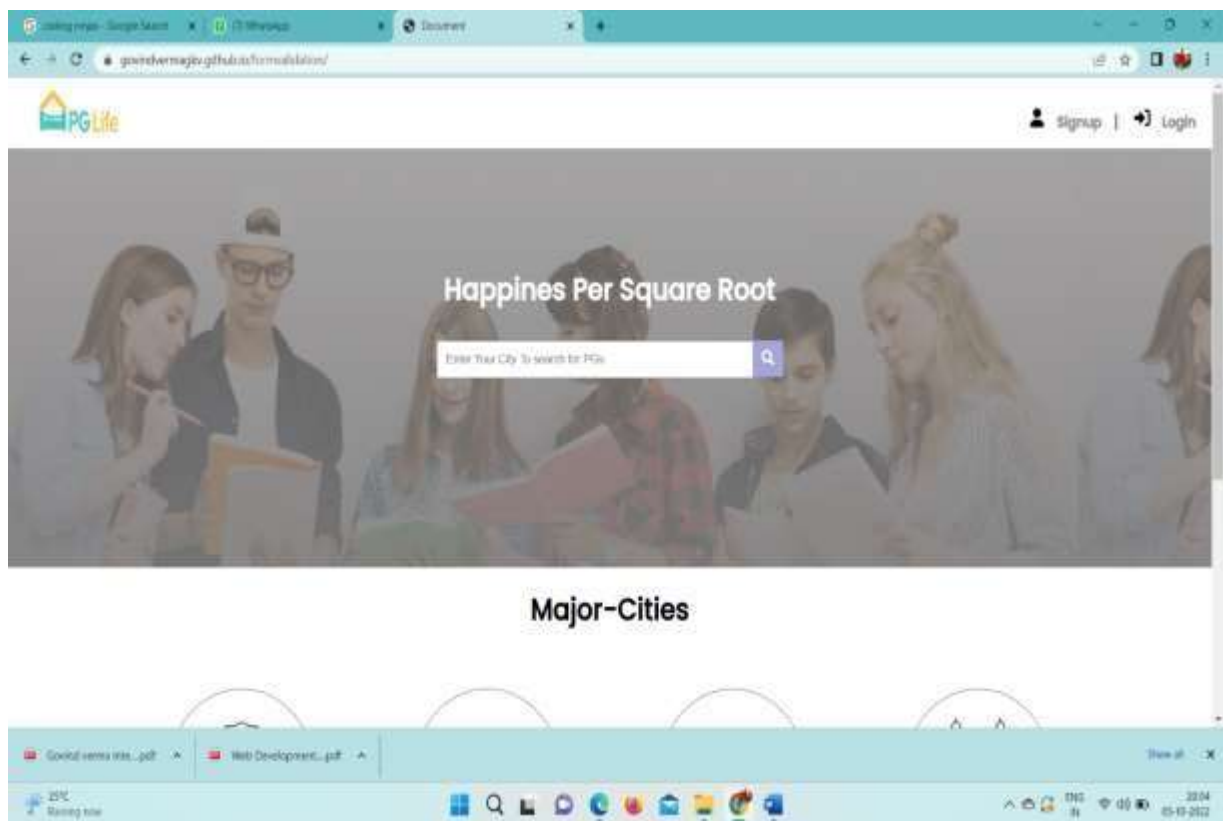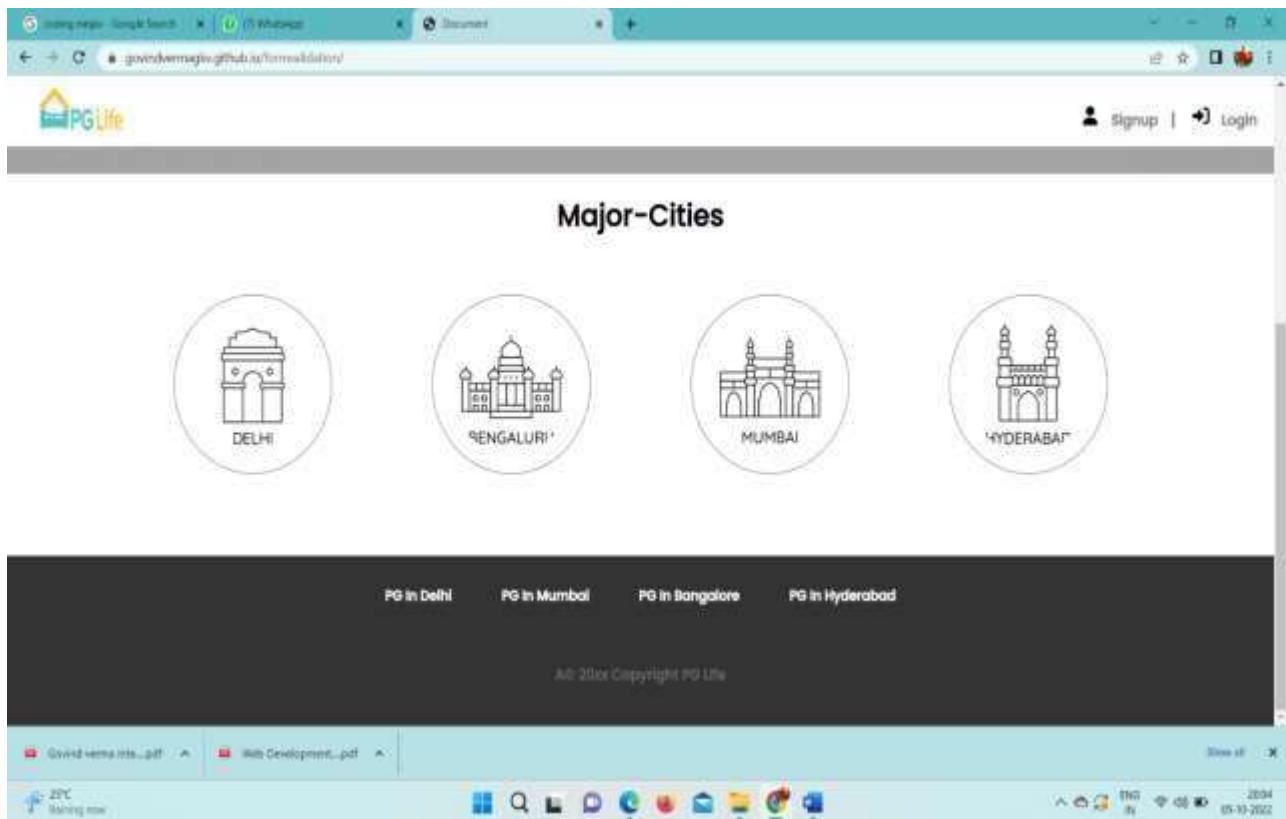
INPUT SCREEN

```
File   Edit   Format   Run   Options   Window   Help
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'main_code.ui'
#'M O T CODE'
# Created by: PyQt5 UI code generator 5.14.2
#
# WARNING! All changes made in this file will be lost!

from calculate_points import player_points
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QMessageBox

from open import Ui_Dialog as Open     # importing open window dialogbox
from newteam import Ui_Dialog as New      # importing new window dialogbox
from evaluate import Ui_evaluate_team as Eva  # importing evaluate window dialog

import sqlite3
fant=sqlite3.connect('cricket_db.db')  # connecting to database file(fandatabase
fantcurs=fant.cursor()


class Ui_MainWindow(object):
    def __init__(self):
        self.newDialog = QtWidgets.QMainWindow()
        self.new_screen = New()
        self.new_screen.setupUi(self.newDialog)

        self.EvaluateWindow = QtWidgets.QMainWindow()
        self.eval_screen = Eva()
        self.eval_screen.setupUi(self.EvaluateWindow)

        self.openDialog = QtWidgets.QMainWindow()
        self.open_screen = Open()
        self.open_screen.setupUi(self.openDialog)

    # FILE OPENING MENU
    def file_open(self):
        self.open_screen.setupUi(self.openDialog)
        self.openDialog.show()
        self.open_screen.openbtn.clicked.connect(self.openteam)
                                                              Ln: 28   Cnk: 0
```

```
Edit   Format   Run   Options   Window   Help
        self.BOWL.setText(str(self.bowlerscount))
        self.BAT.setText(str(self.batsmencount))
        self.ARL.setText(str(self.alrdscount))
        self.WK.setText(str(self.wicketerscount))
        self.list1.clear()
        self.load_names()

        self.sel_player.clear()


    #SAVE TEAM MENU
    def file_save(self):
        if not self.error():    #IF THERE IS AN ERROR
            msg = QMessageBox()
            msg.setIcon(QMessageBox.Critical)
            msg.setInformativeText(' Inufficient Players OR Points !!')
            msg.setWindowTitle("Fantasy Cricket")
            msg.exec_()
        elif self.error():   # IF NO ERROR
            try:
                fantcurs.execute("SELECT DISTINCT name FROM teams;")
                x = fantcurs.fetchall()
                for i in x:
                    if self.team_name.text() == i[0]:     # CHECKING IF THE TEAMN
                        print('Updating already there')
                        fantcurs.execute("DELETE  FROM teams WHERE name='" + se
            except:
                print('error')
            for i in range(self.sel_player.count()):
                # print('----addding--')
                # print('teamname: ',self.team_name.text())
                # print('playername: ',self.list1[i])
                # print('points: ', player_points[self.list1[i]])
                try:
                    fantcurs.execute("INSERT INTO teams (name,players,value) VA
                                     (self.team_name.text(), self.list1[i], pla
                    # self.file_evaluate()
                except:
                    print('error in operation!')
```

,

```
File   Edit   Format   Run   Options   Window   Help

    def error(self):
        msg = QMessageBox()
        if self.wicketerscount > 1:
            msg.setIcon(QMessageBox.Critical)
            # msg.setText("Error")
            msg.setInformativeText('Only 1 wicketkeeper is allowed!')
            msg.setWindowTitle("Error")
            msg.exec_()
            return 0
        elif self.totalcount > 11:
            msg.setIcon(QMessageBox.Critical)
            msg.setInformativeText('No more than 11 players allowed!')
            msg.setWindowTitle("Selection Error")
            msg.exec_()
            return 0
        elif self.totalcount < 11 :
            return 0
        elif self.wicketerscount < 1:
            return 0
        elif self.avail_points <= -1:
            msg.setIcon(QMessageBox.Critical)
            msg.setInformativeText('Not enough points!')
            msg.setWindowTitle("Selection Cricket")
            msg.exec_()
            return 0

        return 1


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

```
File   Edit   Format   Run   Options   Window   Help
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'newteam.ui'
#
# Created by: PyQt5 UI code generator 5.14.2
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(407, 260)
        self.frame = QtWidgets.QFrame(Dialog)
        self.frame.setGeometry(QtCore.QRect(0, -20, 401, 231))
        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame.setObjectName("frame")
        self.label = QtWidgets.QLabel(self.frame)
        self.label.setGeometry(QtCore.QRect(110, 50, 210, 22))
        font = QtGui.QFont()
        font.setFamily("Segoe UI Semibold")
        font.setPointSize(11)
        font.setBold(True)
        font.setItalic(True)
        font.setWeight(75)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.team_name = QtWidgets.QLineEdit(self.frame)
        self.team_name.setGeometry(QtCore.QRect(90, 91, 221, 41))
        self.team_name.setObjectName("team_name")
        self.savename = QtWidgets.QPushButton(self.frame)
        self.savename.setGeometry(QtCore.QRect(140, 160, 93, 28))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(8)
        font.setBold(False)
```

```
File  Edit  Format  Run  Options  Window  Help
            font.setBold(True)
            font.setItalic(True)
            font.setWeight(75)
            self.label.setFont(font)
            self.label.setObjectName("label")
            self.team_name = QtWidgets.QLineEdit(self.frame)
            self.team_name.setGeometry(QtCore.QRect(90, 91, 221, 41))
            self.team_name.setObjectName("team_name")
            self.savename = QtWidgets.QPushButton(self.frame)
            self.savename.setGeometry(QtCore.QRect(140, 160, 93, 28))
            font = QtGui.QFont()
            font.setFamily("MS Sans Serif")
            font.setPointSize(8)
            font.setBold(False)
            font.setItalic(True)
            font.setWeight(50)
            self.savename.setFont(font)
            self.savename.setStyleSheet("font: italic 8pt \"MS Sans Serif\";")
            self.savename.setObjectName("savename")

            self.retranslateUi(Dialog)
            QtCore.QMetaObject.connectSlotsByName(Dialog)

        def retranslateUi(self, Dialog):
            _translate = QtCore.QCoreApplication.translate
            Dialog.setWindowTitle(_translate("Dialog", "new_team"))
            self.label.setText(_translate("Dialog", "Create New Team"))
            self.team_name.setPlaceholderText(_translate("Dialog", "enter team name"
            self.savename.setText(_translate("Dialog", "Save"))


    if __name__ == "__main__":
        import sys
        app = QtWidgets.QApplication(sys.argv)
        Dialog = QtWidgets.QDialog()
        ui = Ui_Dialog()
        ui.setupUi(Dialog)
        Dialog.show()
        sys.exit(app.exec_())
```

```
File  Edit  Format  Run  Options  Window  Help
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'open.ui'
#
# Created by: PyQt5 UI code generator 5.14.2
#
# WARNING! All changes made in this file will be lost!


from PyQt5 import QtCore, QtGui, QtWidgets
import sqlite3
match=sqlite3.connect('cricket_db.db')
matchcur=match.cursor()

class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(368, 274)
        self.label = QtWidgets.QLabel(Dialog)
        self.label.setGeometry(QtCore.QRect(80, 30, 299, 40))
        font = QtGui.QFont()
        font.setFamily("Segoe Print")
        font.setPointSize(14)
        font.setBold(True)
        font.setWeight(75)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.openbtn = QtWidgets.QPushButton(Dialog)
        self.openbtn.setGeometry(QtCore.QRect(120, 160, 93, 49))
        font = QtGui.QFont()
        font.setFamily("Microsoft Sans Serif")
        font.setPointSize(10)
        font.setBold(False)
        font.setItalic(False)
        font.setWeight(50)
        self.openbtn.setFont(font)
        self.openbtn.setStyleSheet("font: 10pt \"Microsoft Sans Serif\";")
        self.openbtn.setObjectName("openbtn")
```

```
ile  Edit  Format  Run  Options  Window  Help
        font = QtGui.QFont()
        font.setFamily("Microsoft Sans Serif")
        font.setPointSize(10)
        font.setBold(False)
        font.setItalic(False)
        font.setWeight(50)
        self.openbtn.setFont(font)
        self.openbtn.setStyleSheet("font: 10pt \"Microsoft Sans Serif\";")
        self.openbtn.setObjectName("openbtn")
        self.open_cb = QtWidgets.QComboBox(Dialog)
        self.open_cb.setGeometry(QtCore.QRect(70, 100, 211, 31))
        self.open_cb.setObjectName("open_cb")

        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)

        teams= matchcur.execute("SELECT DISTINCT name FROM teams;")   # fetching
        y= teams.fetchall()
        for i in y:
            self.open_cb.addItem(i[0])


    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
        self.label.setText(_translate("Dialog", "select team  to open"))
        self.openbtn.setText(_translate("Dialog", "open"))


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Dialog = QtWidgets.QDialog()
    ui = Ui_Dialog()
    ui.setupUi(Dialog)
    Dialog.show()
    sys.exit(app.exec_())
```

```
File  Edit  Format  Run  Options  Window  Help
        font.setBold(False)
        font.setItalic(False)
        font.setWeight(50)
        self.label.setFont(font)
        self.label.setStyleSheet("font: 14pt \"MS Shell Dlg 2\";")
        self.label.setObjectName("label")
        self.finalscore = QtWidgets.QLabel(Dialog)
        self.finalscore.setGeometry(QtCore.QRect(110, 120, 131, 41))
        font = QtGui.QFont()
        font.setFamily("MS Shell Dlg 2")
        font.setPointSize(18)
        font.setBold(False)
        font.setItalic(False)
        font.setWeight(50)
        self.finalscore.setFont(font)
        self.finalscore.setStyleSheet("color: rgb(129, 129, 193);\n"
"\n"
"font: 18pt \"MS Shell Dlg 2\";")
        self.finalscore.setAlignment(QtCore.Qt.AlignCenter)
        self.finalscore.setObjectName("finalscore")

        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)

    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
        self.label.setText(_translate("Dialog", "Your Team Score :"))
        self.finalscore.setText(_translate("Dialog", "0"))


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Dialog = QtWidgets.QDialog()
    ui = Ui_Dialog()
    ui.setupUi(Dialog)
    Dialog.show()
    sys.exit(app.exec_())
```

3. Requirement Analysis

Problem Definition

Problem 1: High cost.

The way we used to work, we couldn't tell you what any given project would cost. One part of the problem is that we don't know what you want to buy, or how much detailed work would be necessary before declaring victory and calling the project complete.

Problem2: PRODUCT DEFINATION

It is an game where you create a team of real cricket players and score points depending on how your chosen players perform in real life matches. To win a tournament, you must try and get the maximum points and the No. 1 rank amongst other participants.

Problem3: FEASIBILITY ANALYSIS

I am building an software for gaming purposes using an specific technology named python. It is a game software where you can create virtual team according to your choice and score points to win an tournament.

This software is created for motivating street cricket and adding more fun and entertainment to cricket. The   components that are used in this demo can be integrated to a high extent to provide statics to different components of cricket. This project helps in providing real time on field actions there by helping its user of the current actions happening on field.

Sample of Rules

Batting
- 1 point for 2 runs scored
- Additional 5 points for half century
- Additional 10 points for century
- 2 points for strike rate (runs/balls faced) of 80-100
- Additional 4 points for strike rate>100
- 1 point for hitting a boundary (four) and 2 points for over boundary (six)

Bowling
- 10 points for each wicket
- Additional 5 points for three wickets per innings
- Additional 10 points for 5 wickets or more in innings

Planning and scheduling (Gantt chart)

Gantt chart-
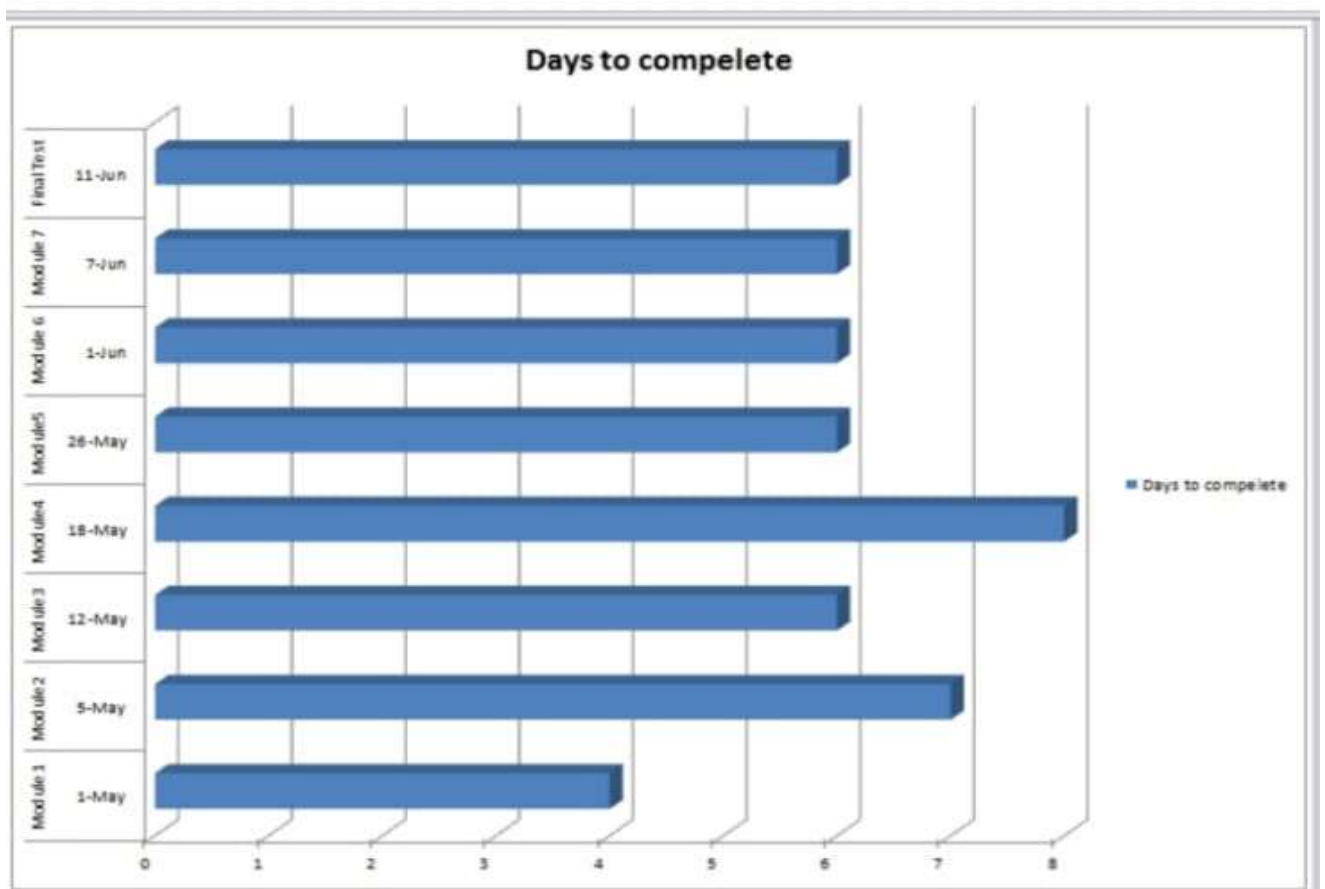
Pert chart-

Hardware Requirement
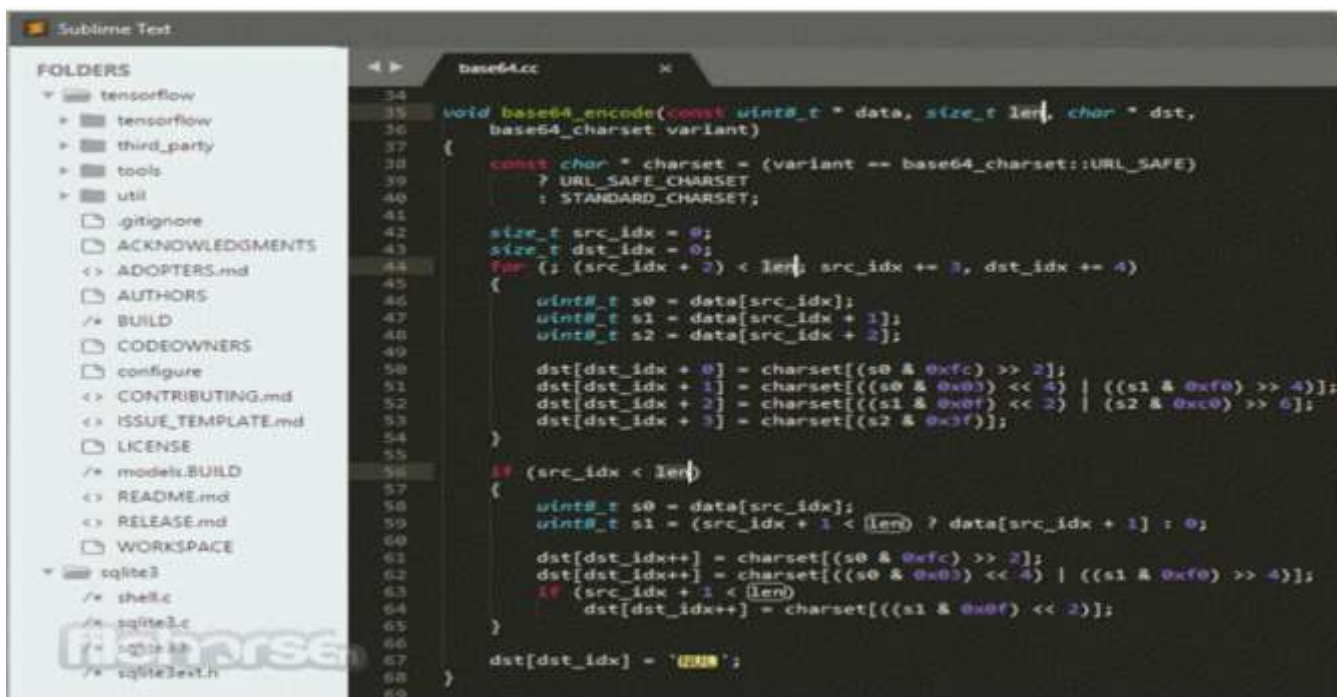
	Windows requirements	**Mac requirements**	**Linux requirements**
Operating system	Windows 8 or later	macOS Sierra 10.12 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Processor	Intel Pentium 4 or	Intel later	Intel Pentium 4 or later
Memory	2 GB minimum, 4 GB recommended		
Screen resolution	1280x1024 or larger		
Application window size	1024x680 or larger		
Internet connection	Required		

Hardware requirements are the configuration that a system must have for a hardware or software application **to run smoothly and efficiently**. hardware requirements are **the requirements of a hardware device**. Most hardware only has operating system requirements or compatibility. In my training, these are the hardware requirements to make responsive web applications.

Software Requirement

Run Python in the Integrated Development environment(IDE)

.

Here is a screenshot of the online text editor:-

4. Preliminary module description

Responsive Web design is the approach that suggests that design and development should respond to the user's behavior and environment based on screen size, platform, and orientation.
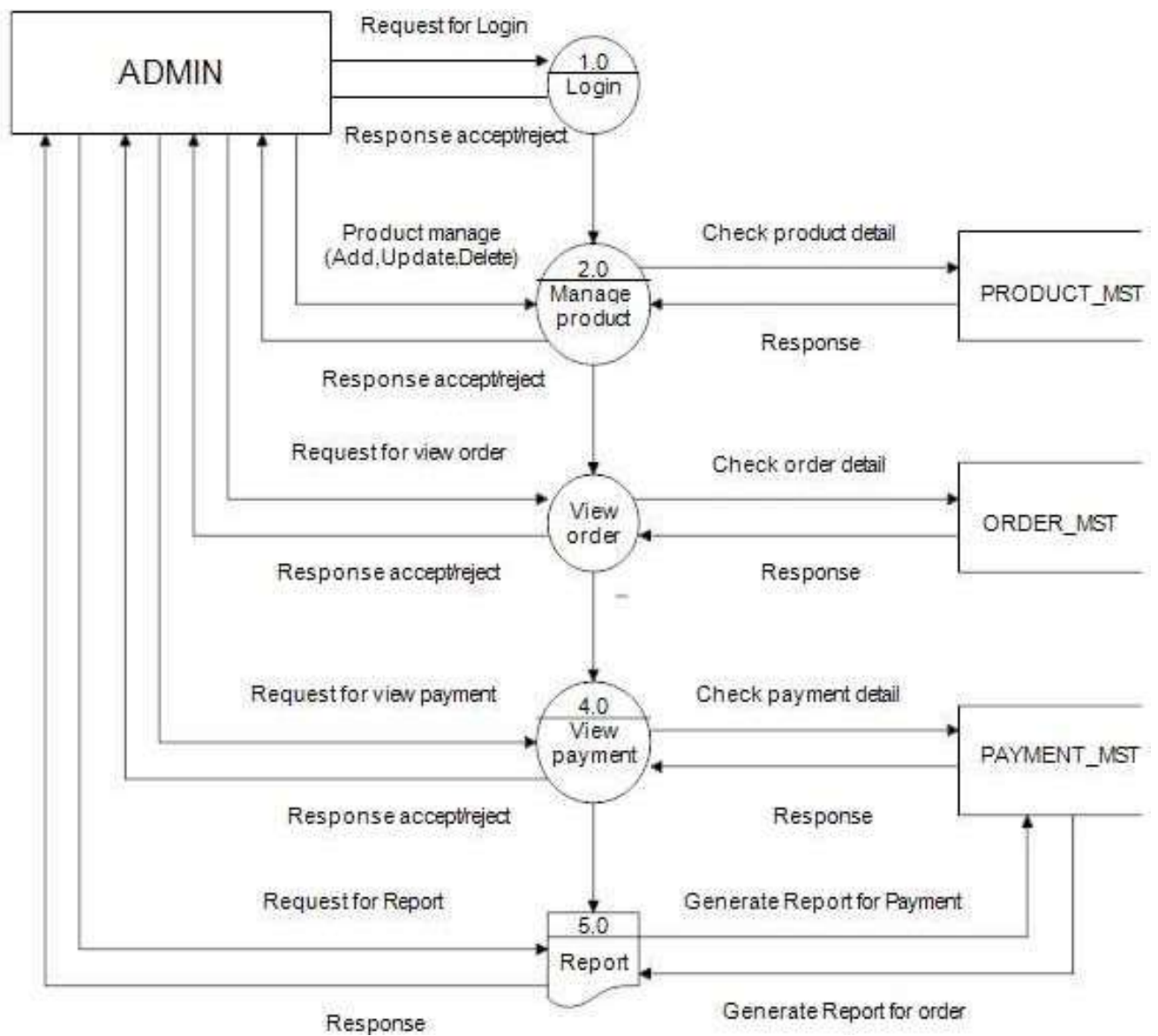
The practice consists of a mix of flexible grids and layouts, images, and intelligent use of CSS media queries. As the user switches from their laptop to iPad, the website should automatically switch to accommodate for resolution, image size, and scripting abilities.
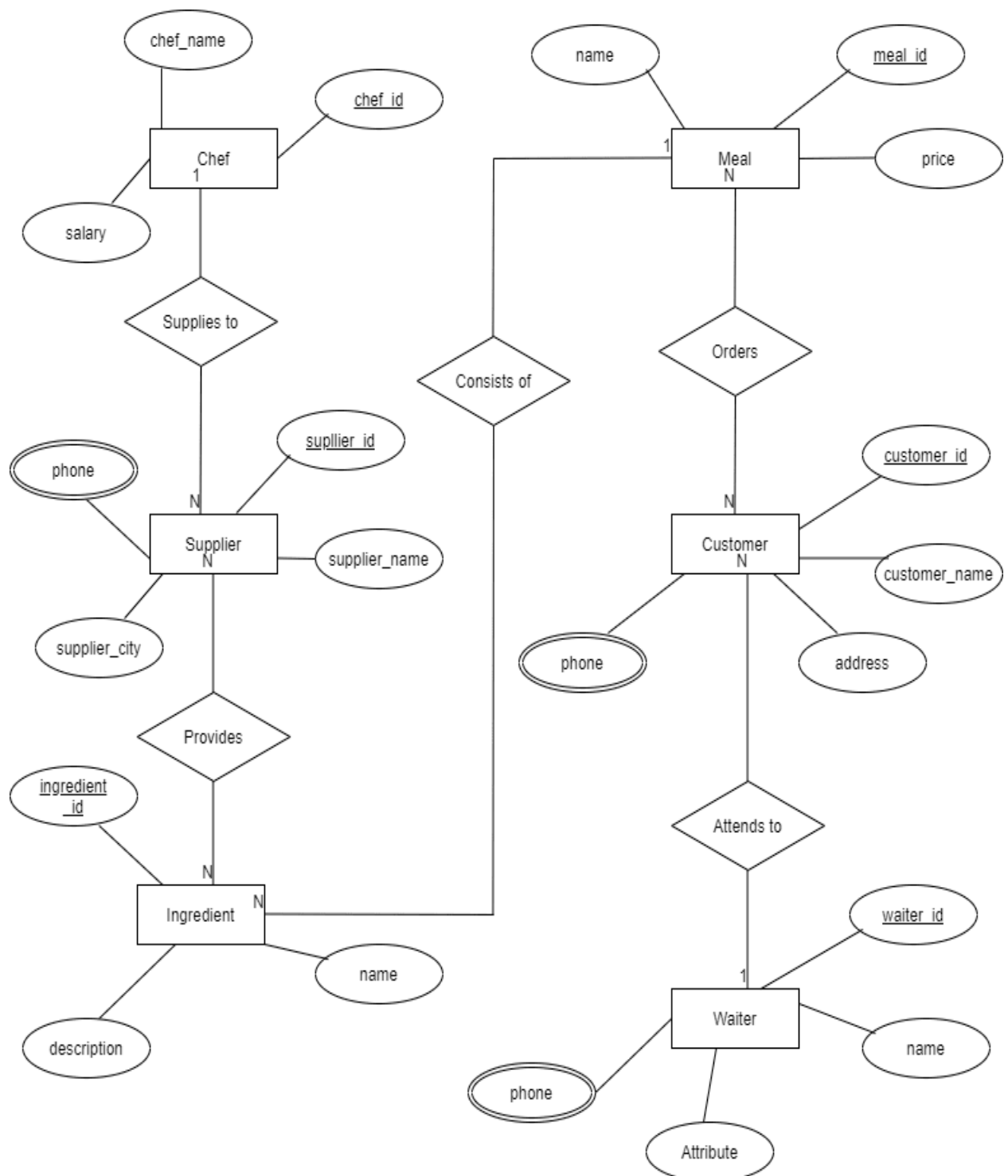
A Responsive web design **automatically adjusts for different-sized screens and viewports**. With a responsive website, someone can browse your website from any device and it will still look and function perfectly.

- Responsive web design has become a hot topic among web designers who want to provide a fair and consistent experience across all devices. • Responsive web design is divided into three main components: the media query, the web browser, and the responsive web interface itself.
- Modules are **common website components that are used like building blocks to create pages**. Each module is a collection of "generic" elements — images, text, buttons, etc. Many modules are combined to build a page. This breaks complex problems into smaller components that are easier to understand, communicate and build.
- A responsive web design **automatically adjusts for different-sized screens and viewports**. With a responsive website, someone can browse your website from any device and it will still look and function perfectly.

- A responsive web app refers to **a design that responds effectively to the behavior of the users and the environment depending on the size of the screen, orientation, and platform**. A responsive web app design has a wide range of flexible layouts, grids, and images.
  Responsive web design is **the practice of building a website suitable to work on every device and every screen size, no matter how large or small, mobile or desktop**. It is a principle that focuses on optimizing a website so that it is flexible, adaptive, and provides a superior user experience.

5. System Designing

DFD

ERD

6.References:

- https://trainings.internshala.com/python-training

- https://www.w3schools.com/python/

- https://wiki.python.org/moin/PyQt/Tutorials

- https://www.tutorialspoint.com/pyqt/

- https://www.tutorialspoint.com/sqlite/sqlite_quick_guide.htm

- https://www.quora.com