****

**BHARATIYA VIDYA BHAVAN**

**SENIOR SECONDARY SCHOOL**

**PERUMTHIRUTHI, ELATHUR (P.O)**

**KOZHIKODE – 673303**

**CBSE AFFILIATION NUMBER : 93091**

**COMPUTER SCIENCE**

**ALL INDIA SENIOR SECONDARY SCHOOL CERTIFICATE EXAMINATION**

*This is to certify that …Vaishnav M Santhosh… of class …12 A… has satisfactorily completed the investigatory project prescribed by the Central Board of Secondary Education for the …Computer Science……. Course in the laboratory of this school in the year …2020-21…*

Date : ………………………

Name of the Candidate : …Vaishnav M Santhosh… Signature of the teacher in charge of batch

Reg. No : …………………….

Date of the Practical Examination : …………………

Internal Examiner :

External Examiner : Signature of the Principal

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS [ T O C ]** | | |
| **SER** | **DESCRIPTION** | **PAGE NO** |
| 01 | ACKNOWLEDGEMENT | **4** |
| 02 | ABOUT PYTHON | **5** |
| 03 | INTRODUCTION | **7** |
| 04 | OBJECTIVES OF THE PROJECT | **7** |
| 05 | PROPOSED SYSTEM | **8** |
| 06 | SYSTEM DEVELOPMENT LIFE CYCLE (SDLC) | **9** |
| 07 | PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE | **10** |
| 08 | SOURCE CODE | **14** |
| 09 | TESTING | **26** |
| 10 | OUTPUT | **28** |
| 11 | HARDWARE AND SOFTWARE REQUIREMENTS | **34** |
| 12 | BIBLIOGRAPHY | **36** |

**ACKNOWLEDGEMENT**

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express deep sense of gratitude to almighty God for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level, who continues to look after me despite my flaws,

I express my deep sense of gratitude to **Mrs. Sreeja Unnikrishnan**, The Principal, Bharatiya Vidya Bhavan School Perumthiruthi who has been continuously motivating and extending their helping hand to us.

My sincere thanks to my Computer Science teacher **Mrs.Rajasree V K**, who critically reviewed my project and helped in solving the problems, occurred during implementation of the project.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

### What is Python?

### Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

### It is used for:

### web development (server-side),

### software development,

### mathematics,

### system scripting.

### What can Python do?

### Python can be used on a server to create web applications.

### Python can be used alongside software to create workflows.

### Python can connect to database systems. It can also read and modify files.

### Python can be used to handle big data and perform complex mathematics.

### Python can be used for rapid prototyping, or for production-ready software development.

### Why Python?

### Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

### Python has a simple syntax similar to the English language.

### Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

### Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

### Python can be treated in a procedural way, an object-oriented way or a functional way.

### Good to know

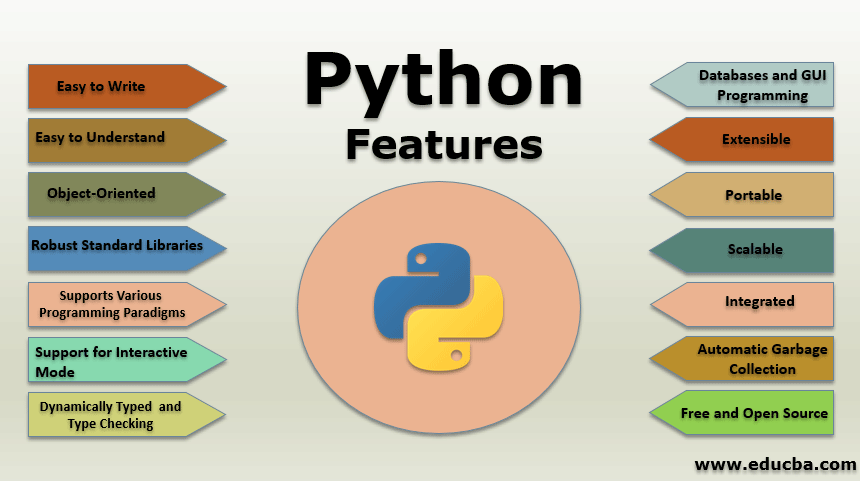
### The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

### It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Spyder,IDLE, PyDev which are particularly useful when managing larger collections of Python files.

### Python Syntax compared to other programming languages

* Python was designed for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**FEATURES OF PYTHON**



**PROJECT ON: Gaming Center DBMS**

**INTRODUCTION**

This project is made for Gaming stores as an automation initiative. Code has been complied in PyCharm IDE. This is a End-User Friendly Interface with slight admin Rights for the shop admins. Also isn’t that much of a resource hog as well, just 400 lines of code. Also the passwords are encrypted by AES encryption.

**OBJECTIVES OF THE PROJECT**

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

**PROPOSED SYSTEM**

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today’s merciless competition where not to wise saying **“to err is human”** no longer valid, it’s out dated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organisations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially had to maintain a lot of ledgers and a lot of paperwork has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done.

This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it’s an age of computers of and automating such an organization gives the better look.

**SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)**

****

The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved.

For example, initial project activities might be designated as request, requirements-definition, and planning phases, or initiation, concept-development, and planning phases. End users of the system under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

**PICTORIAL REPRESENTATION OF SDLC:**

******

**PLANNING PHASE**

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively.

A critical part of a project manager’s job is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.

**REQUIREMENTS ANALYSIS PHASE**

The purposes of this phase are to:

* Further define and refine the functional and data requirements and document them in the Requirements Document,
* Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),
* Develop detailed data and process models (system inputs, outputs, and the process.
* Develop the test and evaluation requirements that will be used to determine acceptable system performance**.**

**DESIGN PHASE**

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk. These include:

* Identifying potential risks and defining mitigating design features.
* Performing a security risk assessment.
* Developing a conversion plan to migrate current data to the new system.
* Determining the operating environment.
* Defining major subsystems and their inputs and outputs.
* Allocating processes to resources.
* Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.
* Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.

**DEVELOPMENT PHASE**

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The Development phase consists of:

* Translating the detailed requirements and design into system components.
* Testing individual elements (units) for usability.
* Preparing for integration and testing of the IT system.

**INTEGRATION AND TEST PHASE**

* Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff access the system security and issue a security certification and accreditation prior to installation/implementation.

***Multiple levels of testing are performed, including***:

* Testing at the development facility by the contractor and possibly supported by end users
* Testing as a deployed system with end users working together with contract personnel
* Operational testing by the end user alone performing all functions. Requirements are traced throughout testing,a final Independent Verification & Validation evaluation is performed and all documentation is reviewedand accepted prior to acceptance of the system.

**IMPLEMENTATION PHASE**

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

**OPERATIONS AND MAINTENANCE PHASE**

The system operation is on-going. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization’s needs. When modifications or changes are identified, the system may re-enter the planning phase.

***The purpose of this phase is to:***

* Operate, maintain, and enhance the system.
* Certify that the system can process sensitive information.
* Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
* Determine when the system needs to be modernized, replaced, or retired.

**SOURCE CODE**

**#defines**

***import mysql.connector as mycon***

***import datetime***

***con = mycon.connect(host='127.0.0.1', user='root', password='admin')***

***if con.is\_connected():***

***print("Database Online\n")***

***cur = con.cursor()***

***cur.execute("create database if not exists Infinix")***

***cur.execute("use Infinix")***

***cur.execute("create table if not exists Main(UID int(4) primary key not null, PASSWORD blob, NAME varchar(40) , TIME\_OF\_CREATION varchar(60))")***

***def LoginUser(uid, passw, name):***

***cur.execute("select \* from Main where UID = {}".format(uid))***

***if cur.fetchall() == []:***

***date = datetime.datetime.now()***

***cur.execute("insert into main values({},AES\_ENCRYPT('{}','admin'),'{}', '{}')".format(uid, passw, name, date))***

***print("New Login Created.\nWelcome user {}: {}".format(uid, name))***

***print("Your UID is {}".format(uid))***

***con.commit()***

***status = True***

***else:***

***status = False***

***cur.execute("select AES\_DECRYPT(password, 'admin') from Main where uid = {}".format(uid))***

***t = [item[0] for item in cur.fetchall()]***

***for i in range(3, 0, -1):***

***if passw != t[0]:***

***print("Sorry Wrong Password.{} attempt(s) remaining".format(i-1))***

***passw = input("Enter Password:")***

***elif passw == t[0]:***

***print("###Login Success###")***

***status = True***

***break***

***elif i < 2:***

***print("Sorry out of attempts Shutting Down!")***

***return status***

***def ForgotPass(uid, passnew):***

***promp = input("Are u an admin?(y/n): ")***

***if promp == 'y':***

***root = input("Enter Root Password(Case Sensitive): ")***

***if root == 'Call911':***

***ask = input("Are you sure to change pass of user {}?(y/n):".format(uid))***

***if ask == 'y':***

***cur.execute("update main set password = AES\_ENCRYPT('{}','admin') where uid = {}".format(passnew,uid))***

***print("Successfully changed Password of user", uid)***

***con.commit()***

***else:***

***print("Okei Nvm that")***

***else:***

***print("Sorry You are not authorized to perform this action")***

***else:***

***print("Contact an administrator for help")***

***def Check\_Creation(uid):***

***cur.execute("select TIME\_OF\_CREATION from main where uid = {}".format(uid))***

***date = cur.fetchall()***

***print([item[0] for item in date][0])***

***if not [item[0] for item in date][0]:***

***print("Enter a valid UID")***

***def GameAdd(console):***

***if console.lower() == 'pc':***

***name = input("Game: ")***

***rate = int(input("Rate: "))***

***genre = input("Genre: ")***

***avail = int(input("Availability: "))***

***date = input("Date of Update:")***

***multi = input("Multiplayer? ")***

***ps4 = 'NO'***

***pc = 'yes'***

***cur.execute("use Infinix")***

***cur.execute("Insert into Games values('{}',{},'{}',{},'{}','{}','{}','{}')".format(name, rate, genre, avail,***

***date, multi, ps4, pc))***

***con.commit()***

***print("Added Successfully")***

***elif console.lower() == 'ps4':***

***name = input("Game: ")***

***rate = int(input("Rate: "))***

***genre = input("Genre: ")***

***avail = int(input("Availability: "))***

***date = input("Date of Update:")***

***multi = input("Multiplayer? ")***

***ps4 = 'Yes'***

***pc = 'no'***

***cur.execute("use Infinix")***

***cur.execute(***

***"Insert into Games values('{}',{},'{}',{},'{}','{}','{}', '{}')".format(name, rate, genre, avail, date,***

***multi, ps4, pc))***

***con.commit()***

***print("Added Successfully")***

***else:***

***print("We have only PS4 and PC right now")***

***def GameList(console):***

***if console.lower() == 'pc':***

***n = 1***

***cur.execute("select NAME from games where PC = 'YES'")***

***for i in [item[0] for item in cur.fetchall()]:***

***print(n, '.', i)***

***n += 1***

***elif console.lower() == 'ps4':***

***n = 1***

***cur.execute("select NAME from games where PS4 = 'YES'")***

***for i in [item[0] for item in cur.fetchall()]:***

***print(n, '.', i)***

***n += 1***

***else:***

***print("Enter a valid console Budd\n")***

***pass***

***def GameSearch(Name):***

***cur.execute("select name from games where name like '%{}%'".format(Name))***

***name = [item[0] for item in cur.fetchall()][0]***

***cur.execute("select \* from Games where Name = '{}'".format(name))***

***op = cur.fetchall()***

***if op == []:***

***print("Sorry mate, we don't have {} right now".format(Name))***

***else:***

***print("Name:", [item[0] for item in op][0], '\nRate:', [item[1] for item in op][0], '\nAVAILABLE:',***

***[item[3] for item in op][0])***

***def GameStart(uid, Name, time):***

***cur.execute("create table if not exists Stock\_atm(Name varchar(60) Primary key,Availability int(2))")***

***cur.execute("create table if not exists console\_check(C\_ID int(2) Primary Key, Status int(2) default 1 check(status = 0 or status = 1),UID int(4))")***

***cur.execute("create table if not exists user\_check(UID int(4) Primary key, Status int(2) default 0 check(Status = 0 or Status = 1),Pending\_Payment int(5) default 0)")***

***cur.execute("insert into user\_check(UID) select UID from main where not exists (select UID from user\_check where user\_check.uid = main.uid)")***

***try:***

***cur.execute('insert into Stock\_atm select Name, availability from Games')***

***for i in range(10):***

***cur.execute("insert into console\_check(C\_ID,UID) value({},{})".format(i + 1, uid))***

***con.commit()***

***except:***

***pass***

***con.commit()***

***cur.execute("select \* from stock\_atm where Name = '{}'".format(Name))***

***op = cur.fetchall()***

***stock = [item[1] for item in op][0]***

***cur.execute("select C\_ID from console\_check where Status = 1")***

***c\_all = cur.fetchall()***

***c\_id = [item[0] for item in c\_all][0]***

***cur.execute("select status from user\_check where UID = {}".format(uid))***

***status = cur.fetchall()***

***online = [item[0] for item in status][0]***

***cur.execute("select Rate\_Per\_Hour from games where name = '{}'".format(Name))***

***deta = cur.fetchall()***

***rate = [item[0] for item in deta][0]***

***cost = rate \* time***

***if stock != 0 and online == 0:***

***print("Your Game is Ready to Begin")***

***stock -= 1***

***print("You may use Console {}".format(c\_id))***

***status = 0***

***cur.execute("update stock\_atm set availability = {} where Name = '{}'".format(stock, Name))***

***cur.execute("update Console\_check set Status = {} where C\_ID = {}".format(status, c\_id))***

***cur.execute("update user\_check set status = 1, Pending\_Payment = Pending\_Payment + {} where UID = {}".format(cost, uid))***

***con.commit()***

***elif stock == 0:***

***print("Sorry someone is already playing {}".format(Name))***

***elif online == 1:***

***print("You are already playing a game elsewhere. Please Logout before proceeding :D")***

***else:***

***print("An unexpected error occurred, Please contact admin ")***

***def GameStop(uid, name):***

***cur.execute("select status from user\_check where UID = {}".format(uid))***

***status = cur.fetchall()***

***online = [item[0] for item in status][0]***

***if online == 1:***

***cur.execute("select name from games where name like '%{}%'".format(name))***

***deta = cur.fetchall()***

***Name = [item[0] for item in deta][0]***

***cur.execute("update user\_check set status = 0 where UID = {}".format(uid))***

***cur.execute("update console\_check set status = 1 where uid = {}".format(uid))***

***cur.execute("update stock\_atm set availability = availability + 1 where name = '{}'".format(Name))***

***con.commit()***

***print("You have stopped playing {}".format(Name))***

***else:***

***print("You aren't playing any game now")***

***def Logout(uid, passw):***

***cur.execute("select AES\_DECRYPT(password, 'admin') from Main where uid = {}".format(uid))***

***t = [item[0] for item in cur.fetchall()]***

***cur.execute("select Pending\_payment from user\_check where uid = {}".format(uid))***

***payment = [item[0] for item in cur.fetchall()][0]***

***if passw == t:***

***y = input("Are you Sure to logout?(Y/N):")***

***verify = True***

***elif passw != t:***

***print("Sorry wrong password mate;-;")***

***ye = 'n'***

***verify = False***

***if ye.lower() == 'y' and verify and payment == 0:***

***cur.execute("update table console\_check set status = 1 where UID = {}".format(uid))***

***cur.execute("update table user\_check set status = 0 where UID = {}".format(uid))***

***cur.execute("update table log\_check set LOGGED\_IN = 0 where UID = {}".format(uid))***

***cur.execute("update stock\_atm set availability = availability + 1 where name = {}".format(uid))***

***print("Logout Success\nC ya later gator ;)")***

***elif payment != 0:***

***print("Make ur Payment before logging out,STUFF ain't free dude")***

***else:***

***print("Keep playing🙄")***

***def Check\_bill(uid):***

***cur.execute("select Pending\_payment from user\_check where uid = {}".format(uid))***

***payment = [item[0] for item in cur.fetchall()][0]***

***print('You need to pay {} Rupees before you logout'.format(payment))***

***return payment***

***def Pay\_bill(uid, payment):***

***cur.execute("select Pending\_payment from user\_check where uid = {}".format(uid))***

***amount = [item[0] for item in cur.fetchall()][0]***

***if payment == amount:***

***print("Payment Done(Happy-Kachingg-Noices)\nYou are free to logout or keep playing")***

***cur.execute("update user\_check set Pending\_Payment = 0 where uid = {}".format(uid))***

***con.commit()***

***else:***

***print("Payment not enuff, recheck ur bill or get glasses xD")***

***def uid\_check():***

***while True:***

***try:***

***id = int(input("Enter UID:"))***

***break***

***except:***

***print("Enter a Valid ID!")***

***continue***

***return id***

***def Close(root):***

***if root == 'Call911':***

***return True***

***else:***

***return False***

**#main**

***choice = 0***

***while choice != 9876:***

***print("1.User Login/Create ID\n2.Show available Games\n3.Search for a Game?\n4.Start Gaming!\n5.Check/Pay bill\n6.Stop Gaming\n7.Logout\n8.Admin tier")***

***while True:***

***try:***

***choice = int(input("Enter your Choice:"))***

***break***

***except:***

***print("Enter a Valid Number!")***

***continue***

***cond = False***

***cur.execute("create table if not exists log\_check(UID int(4),LOGGED\_IN int(2) default 0)")***

***if choice == 1:***

***prompt = input("Are you an existing User?(Y/N):")***

***if prompt.lower() == 'y':***

***uid = uid\_check()***

***cur.execute("select \* from Main where UID = {}".format(uid))***

***deta = cur.fetchall()***

***if not deta:***

***print("Sorry No records Match, Create New:")***

***prompt = 'n'***

***if prompt.lower() == 'y':***

***passw = input("Enter your password(CaSe SenSitiVe):")***

***cond = LoginUser(uid, passw, 'default')***

***con.commit()***

***if cond:***

***cur.execute("insert into log\_check(UID) select UID from main where not exists (select UID from log\_check where log\_check.uid = main.uid)")***

***cur.execute("update log\_check set LOGGED\_IN = 1 where uid = {}".format(uid))***

***print("\n")***

***else:***

***cur.execute("select Max(uid) from main")***

***try:***

***uid = [item[0] for item in cur.fetchall()][0] + 1***

***except TypeError:***

***uid = 1***

***name = input("Enter your Name:")***

***passw = input("Enter your password(CaSe SenSitiVe):")***

***cond = LoginUser(uid, passw, name)***

***con.commit()***

***if cond:***

***cur.execute("insert into log\_check(UID) select UID from main where not exists (select UID from log\_check where log\_check.uid = main.uid)")***

***cur.execute("update log\_check set LOGGED\_IN = 1 where uid = {}".format(uid))***

***print("\n")***

***elif choice == 2:***

***console = input("We have either PS4 or PC for u to play as of now\nEnter the console to see list(PS4/PC):")***

***GameList(console)***

***print("\n")***

***elif choice == 3:***

***Name = input("Enter a game's name to Search:")***

***GameSearch(Name)***

***print("\n")***

***if 3 < choice < 8:***

***uid = uid\_check()***

***cur.execute("select LOGGED\_IN from log\_check where uid = {}".format(uid))***

***cond = [item[0] for item in cur.fetchall()][0]***

***print("\n")***

***if cond == 1:***

***if choice == 4:***

***name = input("Enter the game u want to play(Name as in list)")***

***cur.execute("select name from games where name like '%{}%'".format(name))***

***deta = cur.fetchall()***

***Name = [item[0] for item in deta][0]***

***time = int(input("Enter time in Hours you wanna play:"))***

***GameStart(uid, Name, time)***

***print("\n")***

***elif choice == 5:***

***pay = Check\_bill(uid)***

***print("Contact an admin and pay {}".format(pay))***

***print("\n")***

***elif choice == 6:***

***name = input("Name of game you stopped playing:")***

***cur.execute("select name from games where name like '%{}%'".format(name))***

***deta = cur.fetchall()***

***Name = [item[0] for item in deta][0]***

***GameStop(uid, Name)***

***elif choice == 7:***

***passw = input("Enter your password to confirm logout:")***

***Logout(uid, passw)***

***print("\n")***

***else:***

***print("Enter an option from 1-8!")***

***print("\n")***

***else:***

***print("Login First mate!")***

***print("\n")***

***elif choice == 9876:***

***break***

***elif choice == 8:***

***root = input("Enter Root password:")***

***if root == 'Call911':***

***while True:***

***print("1.Forgot Password\n2.Check date and time of account creation\n3.Add new Games\n4.Pay bill\n5.Close User-Interface\n6.Exit admin mode\n")***

***while True:***

***try:***

***pmp = int(input("Enter your Choice:"))***

***break***

***except:***

***print("Enter a Valid Number!")***

***continue***

***if pmp == 1:***

***uid = uid\_check()***

***passnew = input("Enter new pass without showing the admin:")***

***ForgotPass(uid, passnew)***

***print("\n")***

***elif pmp == 2:***

***uid = uid\_check()***

***Check\_Creation(uid)***

***print("\n")***

***elif pmp == 3:***

***console = input("PS4 or PC?")***

***GameAdd(console)***

***print("\n")***

***elif pmp == 4:***

***uid = uid\_check()***

***payment = int(input("Enter money received:"))***

***Pay\_bill(uid, payment)***

***print("\n")***

***elif pmp == 5:***

***conf = input("R u sure?(Y/N)")***

***if conf.lower() == 'y':***

***sure = Close(root)***

***if sure:***

***print("Goodbye")***

***choice = 9876***

***break***

***else:***

***print("OK nvm")***

***print("\n")***

***elif pmp == 6:***

***break***

***else:***

***print("Wrong Pass Mate")***

***pass***

**||End of Code||**

**\*\*\***

**TESTING**

Software Testing can be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

**TESTING METHODS**

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

**BLACK BOX TESTING**

Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

**SPECIFICATION-BASED TESTING**

Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks

**ADVANTAGES AND DISADVANTAGES**

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers don't. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed.

That's why there are situations when a black box tester writes many test cases to check something that can be tested by only one test case, and/or some parts of the back end are not tested at all. Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

**WHITE BOX TESTING**

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these)

***Types of white box testing:-***

The following types of white box testing exist:

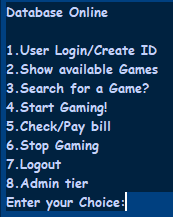
* api testing - Testing of the application using Public and Private APIs.
* Code coverage - creating tests to satisfy some criteria of code coverage.

For example, the test designer can create tests to cause all statements in the program to be executed at least once.

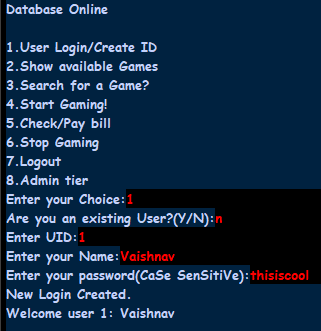
* fault injection methods.
* mutation testing methods.
* static testing - White box testing includes all static testing.

**OUTPUT**

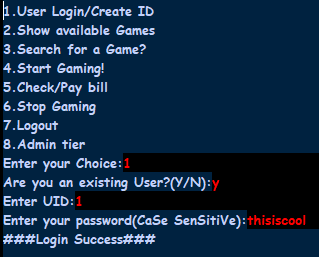
**MAIN PAGE ASKING THE CHOICE OF THE USER**

****

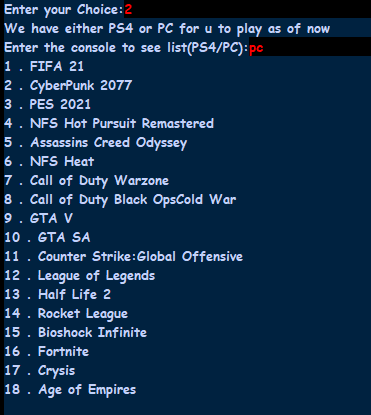
**CREATE LOGIN PAGE**



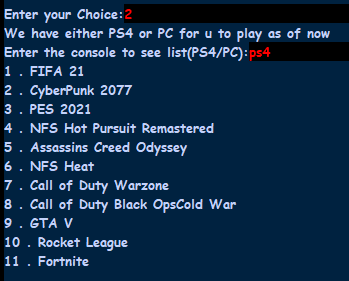
**Returning User Page**

****

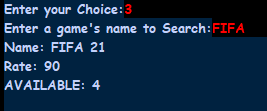
**Show List of Games(PC)**

****

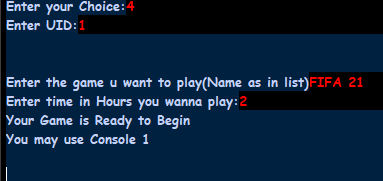
**Show List of Games(PS4)**

****

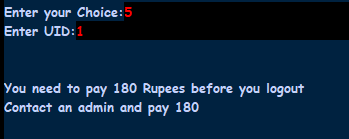
**Game Searching:**

****

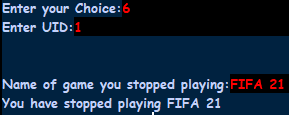
**Start Playing:**

****

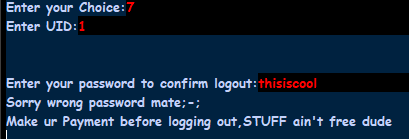
**Check Bill:**

****

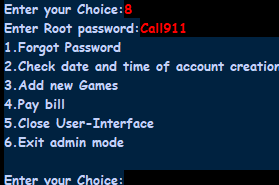
**Stop Playing:**

****

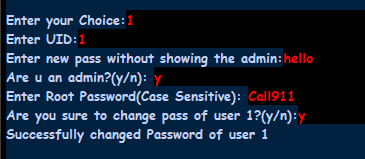
**When user tries to logout before Bill Payment:**

****

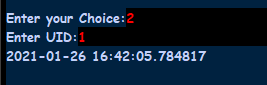
**Admin Tier:**

****

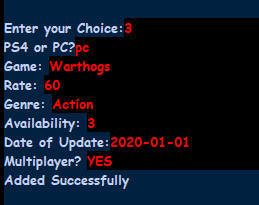
**Password Change:**

****

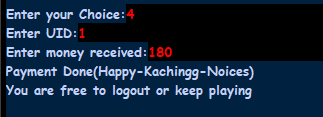
**Check Date and Time of ID Creation:**

****

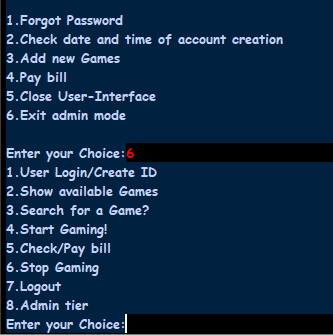
**Add Game:**

****

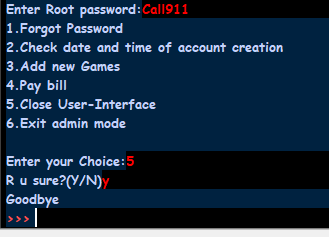
**Bill Pay (In admin-tier):**

****

**Exit Admin Mode:**

****

**Close Interface:**

****

**HARDWARE AND SOFTWARE REQUIREMENTS**

I.OPERATING SYSTEM : WINDOWS 7 AND ABOVE

II. PROCESSOR : Intel Pentium or higher compatible processor

III. RAM : 512MB+

IV. Hard disk : SATA 40 GB OR ABOVE

V. CD/DVD r/w multi drive combo: (If back up required)

VI. MONITOR 14.1 or 15 -17 inch

VII. Key board and mouse

VIII. Printer : (if print is required – [Hard copy])

**SOFTWARE REQUIREMENTS:**

1. Windows OS
2. Python
3. My SQL

.

**Future Enhancement**

**In any project, though present satisfaction is important but also it is equally important to see and visualize the future scope. The project which is developed now may be need to undergo some changes in future in order to match up the technology prevailing that time, thus change due to development in technology are advisable.**

**Some of the future scope are :**

1. **Program can be Deployed for an online DBMS.**
2. **Program as such has limited to no Errors, so application on a large scale is plausible.**

**BIBLIOGRAPHY**

1. ***Computer science With Python - Class XII By : Preeti Arora***
2. ***Sumita Arora Computer Science Text Class XII***
3. ***Website:*** [**https://www3resource.com**](https://www.w3resource.com)
4. [***http://www.python.org/***](http://www.python.org/)
5. ***Google***
6. ***StackOverflow***
7. ***https://www.w3schools.com***

***\*\*\****