

HOTEL MANAGEMENT SYSTEM

PRESENTATION BY

COMPUTER DEPARTMENT

GROUP MEMBERS

|  |  |
| --- | --- |
| NAME | ROLL NO. |
| ANAM ANSARI | 19401 |
| ADITYA CHOUDHARY | 19406 |
| SAMREEN CHOWLKAR | 19407 |
| AYESHA LOLADIA | 19416 |

****

**MAHARASHTRA STATE**

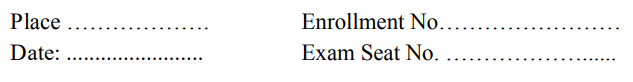
**BOARD OF TECHNICAL EDUCATION**

**CERTIFICATE**

THIS TO CERTIFIY THAT

**Ms. Anam Ansari ..…. 19401**

Of **3rd Year/ 6th Sem** Diploma in **Computer Engineering,** in the Institute of **M. H. Saboo Siddik Polytechnic, Mumbai (0002)** has completed the term work Satisfactory in the subject **Mobile Application Development (22617)** as prescribed by, **Maharashtra State Board of Technical Education, Mumbai**

****

**Subject Teacher Head of the Department Principal**

****

**MAHARASHTRA STATE**

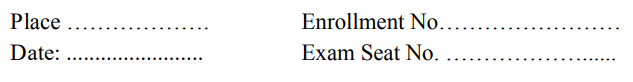
**BOARD OF TECHNICAL EDUCATION**

**CERTIFICATE**

THIS TO CERTIFIY THAT

**Mr. Aditya Choudhary ..…. 19406**

Of **3rd Year/ 6th Sem** Diploma in **Computer Engineering,** in the Institute of **M. H. Saboo Siddik Polytechnic, Mumbai (0002)** has completed the term work Satisfactory in the subject **Mobile Application Development (22617)** as prescribed by, **Maharashtra State Board of Technical Education, Mumbai**

****

**Subject Teacher Head of the Department Principal**

****

**MAHARASHTRA STATE**

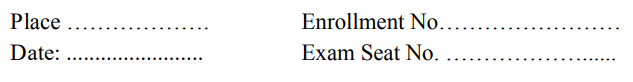
**BOARD OF TECHNICAL EDUCATION**

**CERTIFICATE**

THIS TO CERTIFIY THAT

**Ms. Samreen Chowlkar ..…. 19407**

Of **3rd Year/ 6th Sem** Diploma in **Computer Engineering,** in the Institute of **M. H. Saboo Siddik Polytechnic, Mumbai (0002)** has completed the term work Satisfactory in the subject **Mobile Application Development (22617)** as prescribed by, **Maharashtra State Board of Technical Education, Mumbai**

****

**Subject Teacher Head of the Department Principal**

****

**MAHARASHTRA STATE**

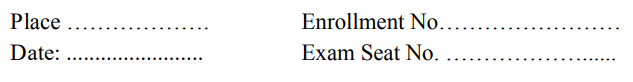
**BOARD OF TECHNICAL EDUCATION**

**CERTIFICATE**

THIS TO CERTIFIY THAT

**Ms. Ayesha Loladia ..…. 19416**

Of **3rd Year/ 6th Sem** Diploma in **Computer Engineering,** in the Institute of **M. H. Saboo Siddik Polytechnic, Mumbai (0002)** has completed the term work Satisfactory in the subject **Mobile Application Development (22617)** as prescribed by, **Maharashtra State Board of Technical Education, Mumbai**

****

**Subject Teacher Head of the Department Principal**

**ANNEXURE**

**Evaluation sheet for the micro-project.**

**Academic year: 2021-22. Name of faculty: Shafaque Julaha**

**Course: Mobile Application Development Course code:22617.**

**Semester: 6th / 3rd Year**

**Title of the project: Hotel Management System**

**COs addressed by the micro-project:**

1. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Major learning outcomes achieved by the students by doing the project:**

**(a) Practical Outcomes:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(b) Unit outcomes in cognitive domain:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**(c) Outcomes in effective domain:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Comments/ suggestions about teamwork/ leadership/ inter-personal communication (if any):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Roll No:** | **Student Name** | **Marks out of 6 for**  **performance in group activity** | **Marks out of 4 for performance in oral or presentation** | **Total out of**  **10** |
| **19401** | **Anam Ansari** |  |  |  |
| **19406** | **Aditya Choudhary** |  |  |  |
| **19407** | **Samreen Chowlkar** |  |  |  |
| **19416** | **Ayesha Loladia** |  |  |  |

**(NAME AND SIGNATURE OF FACULTY)**

**INTRODUCTION**

As technology continues its rapid advance across all industries around the world, hospitality is really beginning to see the benefits. Hotel management software has come a long way in helping hoteliers improve the way their business operates, and there is little doubt as to its transformative impact.

A modern [property management system](https://www.mews.com/en) helps you to streamline administrative systems and processes, as well as boost the company's overall operations. It's easy to see why the vast majority of hotel owners believe an excellent management system is essential for their business.

There are countless advantages to a reliable hotel management software system. Whether it’s time saving on manual tasks or increasing direct bookings, every element of a hospitality system should be working towards the end goals of improving efficiency and enhancing the guest experience.

Hotel management is really about overseeing every operation of the property. This requires knowledge of distribution strategy, finance, customer service, staff management, marketing, and more.

In no way should any of these be treated as ‘set and forget’. Hotel management is about constantly evaluating performance is every facet of the business and making necessary adjustments.

Ultimately effective hotel management will not only ensure your hotel stays in business, but is able to profit and grow over time. Think of the hotel as an ecosystem that will get healthier the better you manage it. As your hotel becomes more successful you can upgrade and charge higher rates, pay staff higher wages, and create an experience that guests want to come back for.

It can take time to get everything right however. There are many skills you’ll already possess but many others you need to learn along the way, or else hire staff that can provide the knowledge for you.

HOTEL MANAGEMENT SYSTEM

* **RATIONALE**

Android application development is one of the rising and growing trend in the industry of mobile. This course examines the principles of mobile application design and covers the necessary concepts which are required to understand mobile based applications and develop Android based Applications in particular. After completing this course students will design and build a variety of real-time Apps using Android.

* **COURSE OUTCOMES**

**a.** Interpret features of Android Operating System.

b. Configure Android Environment and Development tools.

c. Develop rich User Interfaces by using layouts and controls.

d. Use User Interface components for android application development.

e. Create Android application using database.

* **RESOURCES USED**

|  |  |
| --- | --- |
| **SR.NO** | **Equipment’s with broad specification** |
| **1** | **Personal Computer, Core i5, RAM – 8GB, HD – 1TB.** |
| **2** | **Windows 10 (Operating System)** |
| **3** | **Android Studio** |
| **4** | **Microsoft Word , Google(chrome).** |

**LITERATURE REVIEW**

1. **What is Android:**

Android is a [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) based on a modified version of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and other [open source](https://en.wikipedia.org/wiki/Open-source_software) software, designed primarily for [touchscreen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer). Android is developed by a consortium of developers known as the [Open Handset Alliance](https://en.wikipedia.org/wiki/Open_Handset_Alliance), with the main contributor and commercial marketer being [Google](https://en.wikipedia.org/wiki/Google).

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the [first commercial Android device](https://en.wikipedia.org/wiki/HTC_Dream) launched in September 2008. The current stable version is [Android 10](https://en.wikipedia.org/wiki/Android_10), released on September 3, 2019. The core Android source code is known as Android Open Source Project (AOSP), which is primarily licensed under the [Apache License](https://en.wikipedia.org/wiki/Apache_License). This has allowed variants of Android to be developed on a range of other electronics, such as [game consoles](https://en.wikipedia.org/wiki/Video_game_console), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera), [PCs](https://en.wikipedia.org/wiki/Personal_computer) and others, each with a specialized user interface. Some well-known derivatives include [Android TV](https://en.wikipedia.org/wiki/Android_TV) for televisions and [Wear OS](https://en.wikipedia.org/wiki/Wear_OS) for wearables, both developed by Google.

Android's source code has been used as the basis of different ecosystems, most notably that of Google which is associated with a suite of [proprietary software](https://en.wikipedia.org/wiki/Proprietary_software) called [Google Mobile Services](https://en.wikipedia.org/wiki/Google_Mobile_Services) (GMS), that frequently comes pre-installed on said devices. This includes core apps such as [Gmail](https://en.wikipedia.org/wiki/Gmail), the [digital distribution](https://en.wikipedia.org/wiki/Digital_distribution) platform [Google Play](https://en.wikipedia.org/wiki/Google_Play) and associated [Google Play Services](https://en.wikipedia.org/wiki/Google_Play_Services) development platform, and usually apps such as the [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) web browser. These apps are licensed by manufacturers of Android devices certified under standards imposed by Google. Other competing Android ecosystems include [Amazon.com](https://en.wikipedia.org/wiki/Amazon.com" \o "Amazon.com)'s [Fire OS](https://en.wikipedia.org/wiki/Fire_OS), or [LineageOS](https://en.wikipedia.org/wiki/LineageOS" \o "LineageOS). Software distribution is generally offered through proprietary [application stores](https://en.wikipedia.org/wiki/Application_store) like [Google Play Store](https://en.wikipedia.org/wiki/Google_Play_Store) or [Samsung Galaxy Store](https://en.wikipedia.org/wiki/Samsung_Galaxy_Store), or open source platforms like [Aptoide](https://en.wikipedia.org/wiki/Aptoide" \o "Aptoide) or [F-Droid](https://en.wikipedia.org/wiki/F-Droid), which use software packages in the [APK](https://en.wikipedia.org/wiki/Android_application_package) format.

Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion [monthly active users](https://en.wikipedia.org/wiki/Monthly_active_users), the largest [installed base](https://en.wikipedia.org/wiki/Installed_base) of any operating system, and as of January 2020, the Google Play Store features over 2.9 million apps.

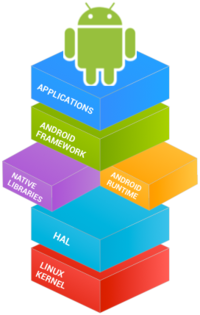
Icon

Description automatically generated

*Fig. Android OS logo*

1. **Android Software Development:**

Android software development is the process by which new applications are created for devices running the [Android operating system](https://en.wikipedia.org/wiki/Android_(operating_system)). Google states that "Android apps can be written using [Kotlin](https://en.wikipedia.org/wiki/Kotlin_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), and [C++](https://en.wikipedia.org/wiki/C%2B%2B) languages" using the Android [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK), while using other languages is also possible. All non-JVM languages, such as [Go](https://en.wikipedia.org/wiki/Go_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [C](https://en.wikipedia.org/wiki/C_(programming_language)), C++ or [assembly](https://en.wikipedia.org/wiki/Assembly_language), need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some programming languages and tools allow cross-platform app support (i.e. for both Android and [iOS](https://en.wikipedia.org/wiki/IOS)). Third party tools, development environments, and language support have also continued to evolve and expand since the initial SDK was released in 2008. In addition, with major business entities like [Walmart](https://en.wikipedia.org/wiki/Walmart), [Amazon](https://en.wikipedia.org/wiki/Amazon_Kindle), and [Bank of America](https://en.wikipedia.org/wiki/Bank_of_America) eyeing to engage and sell through mobiles, mobile application development is witnessing a transformation. The official Android app distribution mechanism to end users is [Google Play](https://en.wikipedia.org/wiki/Google_Play); it also allows staged gradual app release, as well as distribution of pre-release app versions to testers.



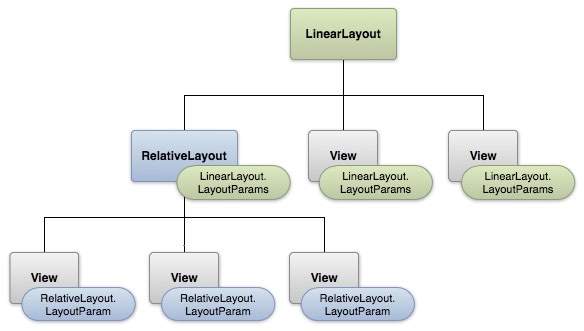
*Fig. Android Software Development*

1. **Android – UI Layouts:**

The basic building block for user interface is a View object which is created from the View class and occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components like buttons, text fields, etc.

The ViewGroup is a subclass of View and provides invisible container that hold other Views or other ViewGroups and define their layout properties.

At third level we have different layouts which are subclasses of ViewGroup class and a typical layout defines the visual structure for an Android user interface and can be created either at run time using View/ViewGroup objects or you can declare your layout using simple XML file main\_layout.xml which is located in the res/layout folder of your project.



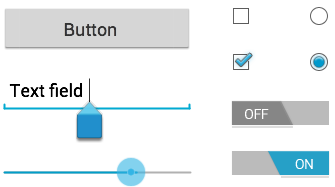
*Fig. Layout params*

There are number of Layouts provided by Android which you will use in almost all the Android applications to provide different view, look and feel.

|  |  |
| --- | --- |
| **Sr.No** | **Layout & Description** |
| 1 | [**Linear Layout**](https://www.tutorialspoint.com/android/android_linear_layout.htm)**:**  LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. |
| 2 | [**Relative Layout**](https://www.tutorialspoint.com/android/android_relative_layout.htm)**:**  RelativeLayout is a view group that displays child views in relative positions. |
| 3 | [**Table Layout**](https://www.tutorialspoint.com/android/android_table_layout.htm)**:**  TableLayout is a view that groups views into rows and columns. |
| 4 | [**Absolute Layout**](https://www.tutorialspoint.com/android/android_absolute_layout.htm)**:**  AbsoluteLayout enables you to specify the exact location of its children. |
| 5 | [**Frame Layout**](https://www.tutorialspoint.com/android/android_frame_layout.htm)  The FrameLayout is a placeholder on screen that you can use to display a single view. |
| 6 | [**List View**](https://www.tutorialspoint.com/android/android_list_view.htm)  ListView is a view group that displays a list of scrollable items. |
| 7 | [**Grid View**](https://www.tutorialspoint.com/android/android_grid_view.htm)  GridView is a ViewGroup that displays items in a two-dimensional, scrollable grid. |

1. **Android – UI Controls:**

Input controls are the interactive components in your app's user interface. Android provides a wide variety of controls you can use in your UI, such as buttons, text fields, seek bars, check box, zoom buttons, toggle buttons, and many more.



*Fig. Android UI Controls*

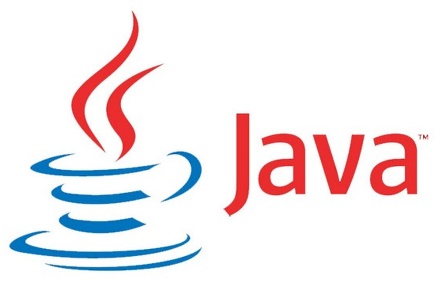
There are number of UI controls provided by Android that allow you to build the graphical user interface for your app.

|  |  |
| --- | --- |
| **Sr.No.** | **UI Control & Description** |
| 1 | [**TextView**](https://www.tutorialspoint.com/android/android_textview_control.htm)**:**  This control is used to display text to the user. |
| 2 | [**EditText**](https://www.tutorialspoint.com/android/android_edittext_control.htm)**:**  EditText is a predefined subclass of TextView that includes rich editing capabilities. |
| 3 | [**AutoCompleteTextView**](https://www.tutorialspoint.com/android/android_autocompletetextview_control.htm)**:**  The AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing. |
| 4 | [**Button**](https://www.tutorialspoint.com/android/android_button_control.htm)**:**  A push-button that can be pressed, or clicked, by the user to perform an action. |
| 5 | [**ImageButton**](https://www.tutorialspoint.com/android/android_imagebutton_control.htm)**:**  An ImageButton is an AbsoluteLayout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user. |
| 6 | [**CheckBox**](https://www.tutorialspoint.com/android/android_checkbox_control.htm)**:**  An on/off switch that can be toggled by the user. You should use check box when presenting users with a group of selectable options that are not mutually exclusive. |
| 7 | [**ToggleButton**](https://www.tutorialspoint.com/android/android_togglebutton_control.htm)**:**  An on/off button with a light indicator. |
| 8 | [**RadioButton**](https://www.tutorialspoint.com/android/android_radiobutton_control.htm)**:**  The RadioButton has two states: either checked or unchecked. |
| 9 | [**RadioGroup**](https://www.tutorialspoint.com/android/android_radiogroup_control.htm)**:**  A RadioGroup is used to group together one or more RadioButtons. |
| 10 | [**ProgressBar**](https://www.tutorialspoint.com/android/android_progressbar.htm)**:**  The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background. |
| 11 | [**Spinner**](https://www.tutorialspoint.com/android/android_spinner_control.htm)**:**  A drop-down list that allows users to select one value from a set. |
| 12 | [**TimePicker**](https://www.tutorialspoint.com/android/android_timepicker_control.htm)**:**  The TimePicker view enables users to select a time of the day, in either 24-hour mode or AM/PM mode. |
| 13 | [**DatePicker**](https://www.tutorialspoint.com/android/android_datepicker_control.htm)**:**  The DatePicker view enables users to select a date of the day. |

1. **Java Programming Language:**

Java is a [general-purpose](https://en.wikipedia.org/wiki/General-purpose_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) that is [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and designed to have as few implementation [dependencies](https://en.wikipedia.org/wiki/Dependency_(computer_science)) as possible. It is intended to let [application developers](https://en.wikipedia.org/wiki/Application_developer) write once, run anywhere (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of the underlying [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture). The [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) of [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) is similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them. As of 2019, Java was one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) according to [GitHub](https://en.wikipedia.org/wiki/GitHub), particularly for [client-server](https://en.wikipedia.org/wiki/Client%E2%80%93server) [web applications](https://en.wikipedia.org/wiki/Web_applications), with a reported 9 million developers.

Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) ([which has since been acquired by Oracle](https://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)). The original and [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) Java [compilers](https://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](https://en.wikipedia.org/wiki/Library_(computing)) were originally released by Sun under [proprietary licenses](https://en.wikipedia.org/wiki/Proprietary_license). As of May 2007, in compliance with the specifications of the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process), Sun had [relicensed](https://en.wikipedia.org/wiki/Software_relicensing) most of its Java technologies under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License). Meanwhile, others have developed alternative implementations of these Sun technologies, such as the [GNU Compiler for Java](https://en.wikipedia.org/wiki/GNU_Compiler_for_Java) (bytecode compiler), [GNU Classpath](https://en.wikipedia.org/wiki/GNU_Classpath) (standard libraries), and [IcedTea](https://en.wikipedia.org/wiki/IcedTea" \o "IcedTea)-Web (browser plugin for applets).

The latest versions are [Java 13](https://en.wikipedia.org/wiki/Java_version_history), released in September 2019, and Java 11, a currently supported [long-term support](https://en.wikipedia.org/wiki/Long-term_support) (LTS) version, released on September 25, 2018; [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation) released for the [legacy](https://en.wikipedia.org/wiki/Legacy_system) [Java 8](https://en.wikipedia.org/wiki/Java_8) LTS the last free public update in January 2019 for commercial use, while it will otherwise still support Java 8 with public updates for personal use up to at least December 2020. Oracle (and others) highly recommend uninstalling older versions of Java because of serious risks due to unresolved security issues. Since Java 9 (and 10 and 12) is no longer supported, Oracle advises its users to immediately transition to the latest version (currently Java 13) or an LTS release.

1. **SQLite Database:**

Input controls are the interactive components in your app's user interface. Android provides a wide variety of controls you can use in your UI, such as buttons, text

SQLite ([/ˌɛsˌkjuːˌɛlˈaɪt/](https://en.wikipedia.org/wiki/Help:IPA/English), [/ˈsiːkwəˌlaɪt/](https://en.wikipedia.org/wiki/Help:IPA/English)) is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) contained in a [C](https://en.wikipedia.org/wiki/C_(programming_language)) [library](https://en.wikipedia.org/wiki/Library_(computer_science)). In contrast to many other database management systems, SQLite is not a [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) database engine. Rather, it is embedded into the end program.

SQLite is [ACID](https://en.wikipedia.org/wiki/ACID)-compliant and implements most of the [SQL](https://en.wikipedia.org/wiki/SQL) standard, generally following [PostgreSQL](https://en.wikipedia.org/wiki/PostgreSQL) syntax. However, SQLite uses a dynamically and weakly [typed](https://en.wikipedia.org/wiki/Data_type) SQL [syntax](https://en.wikipedia.org/wiki/Syntax) that does not guarantee the [domain integrity](https://en.wikipedia.org/wiki/Data_integrity#TYPES). This means that one can, for example, insert a string into a column defined as an integer. SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions, and will store the data as-is if such a conversion is not possible.

SQLite is a popular choice as [embedded database](https://en.wikipedia.org/wiki/Embedded_database) software for local/client storage in [application software](https://en.wikipedia.org/wiki/Application_software) such as [web browsers](https://en.wikipedia.org/wiki/Web_browser). It is arguably the most widely deployed [database engine](https://en.wikipedia.org/wiki/Database_engine), as it is used today by several widespread browsers, [operating systems](https://en.wikipedia.org/wiki/Operating_system), and [embedded systems](https://en.wikipedia.org/wiki/Embedded_system) (such as mobile phones), among others. SQLite has [bindings](https://en.wikipedia.org/wiki/Language_binding) to many programming languages.



*Fig. SQLite logo*

.

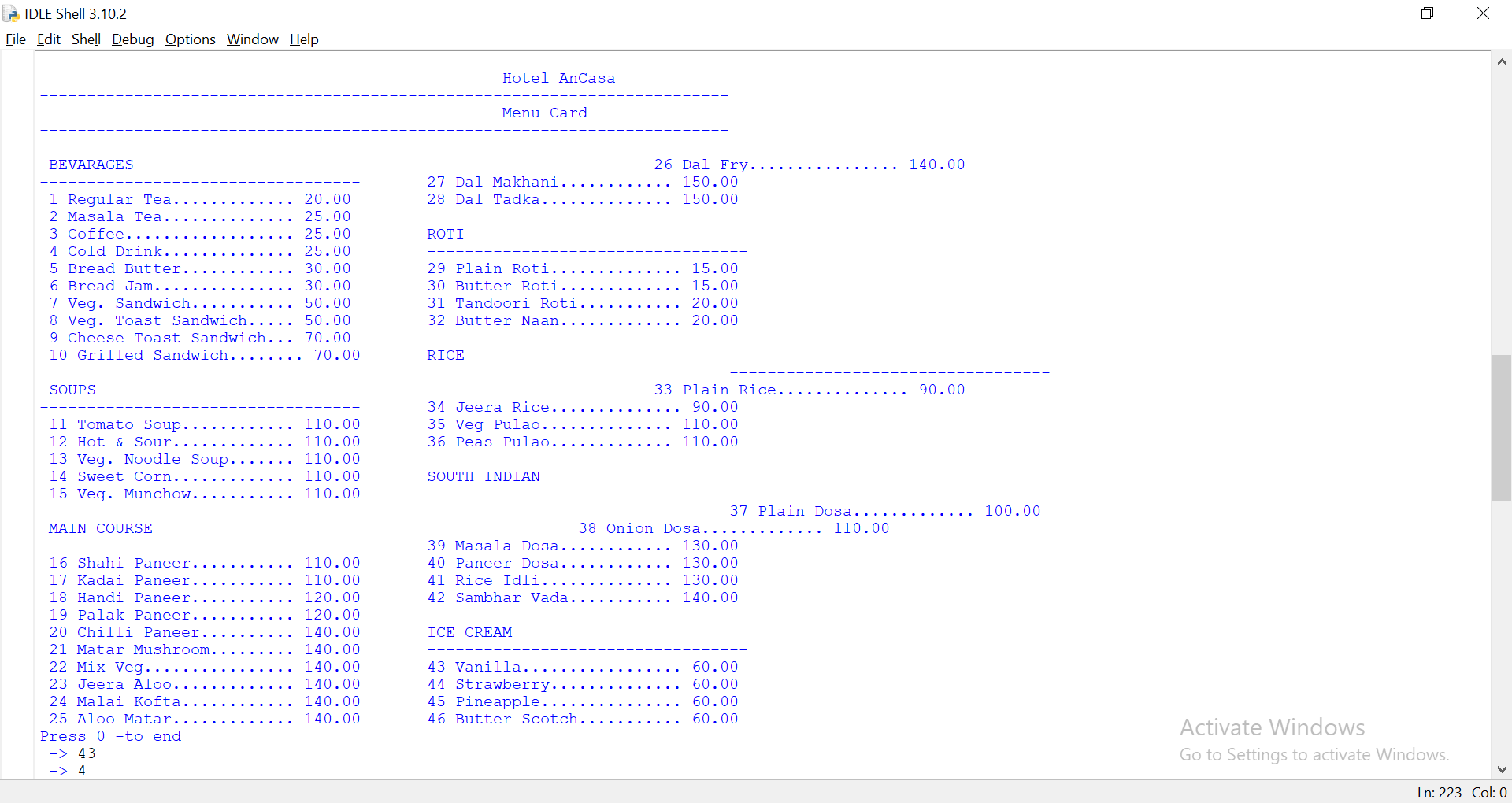
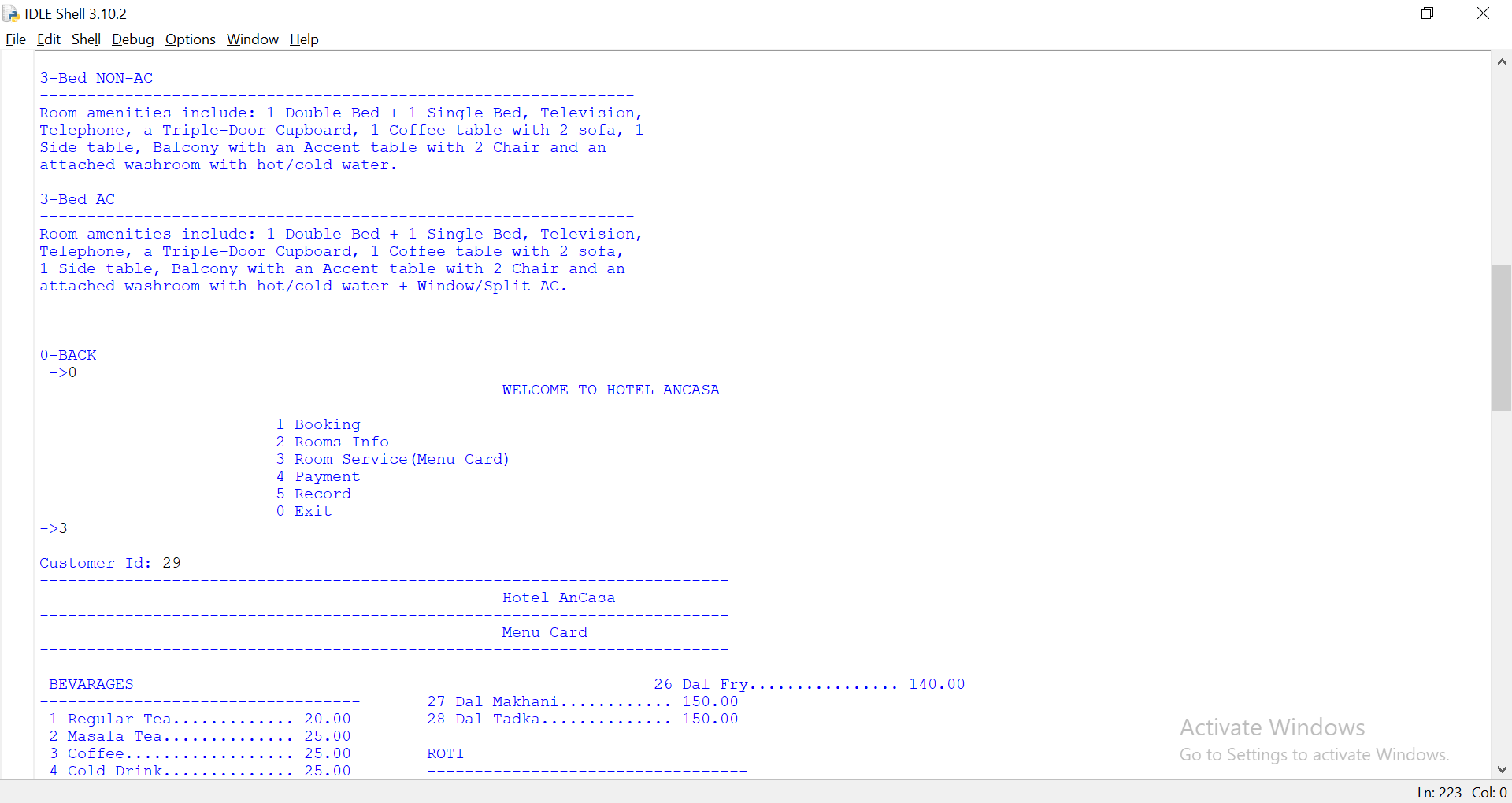
#### **CODE**

import random  
import datetime  
  
# Global List Declaration  
name = []  
phno = []  
add = []  
checkin = []  
checkout = []  
room = []  
price = []  
rc = []  
p = []  
roomno = []  
custid = []  
day = []  
  
# Global Variable Declaration  
  
i = 0  
  
# Home Function  
def Home():  
   
 print("\t\t\t\t\t\t WELCOME TO HOTEL ANCASA\n")  
 print("\t\t\t 1 Booking")  
 print("\t\t\t 2 Rooms Info")  
 print("\t\t\t 3 Room Service(Menu Card)")  
 print("\t\t\t 4 Payment")  
 print("\t\t\t 5 Record")  
 print("\t\t\t 0 Exit")  
  
 ch=int(input("->"))  
   
 if ch == 1:  
 print(" ")  
 Booking()  
   
 elif ch == 2:  
 print(" ")  
 Rooms\_Info()  
   
 elif ch == 3:  
 print(" ")  
 restaurant()  
   
 elif ch == 4:  
 print(" ")  
 Payment()  
   
 elif ch == 5:  
 print(" ")  
 Record()  
   
 else:  
 exit()  
  
# Function used in booking  
  
def date(c):  
   
 if c[2] >= 2022 and c[2] <= 2023:  
   
 if c[1] != 0 and c[1] <= 12:  
   
 if c[1] == 2 and c[0] != 0 and c[0] <= 31:  
   
 if c[2]%4 == 0 and c[0] <= 29:  
 pass  
   
 elif c[0]<29:  
 pass  
   
 else:  
 print("Invalid date\n")  
 name.pop(i)  
 phno.pop(i)  
 add.pop(i)  
 checkin.pop(i)  
 checkout.pop(i)  
 Booking()  
   
   
 # if month is odd & less than equal  
 # to 7th month  
 elif c[1] <= 7 and c[1]%2 != 0 and c[0] <= 31:  
 pass  
   
 # if month is even & less than equal to 7th  
 # month and not 2nd month  
 elif c[1] <= 7 and c[1]%2 == 0 and c[0] <= 30 and c[1] != 2:  
 pass  
   
 # if month is even & greater than equal  
 # to 8th month  
 elif c[1] >= 8 and c[1]%2 == 0 and c[0] <= 31:  
 pass  
   
 # if month is odd & greater than equal  
 # to 8th month  
 elif c[1]>=8 and c[1]%2!=0 and c[0]<=30:  
 pass  
   
 else:  
 print("Invalid date\n")  
 name.pop(i)  
 phno.pop(i)  
 add.pop(i)  
 checkin.pop(i)  
 checkout.pop(i)  
 Booking()  
   
 else:  
 print("Invalid date\n")  
 name.pop(i)  
 phno.pop(i)  
 add.pop(i)  
 checkin.pop(i)  
 checkout.pop(i)  
 Booking()  
   
 else:  
 print("Invalid date\n")  
 name.pop(i)  
 phno.pop(i)  
 add.pop(i)  
 checkin.pop(i)  
 checkout.pop(i)  
 Booking()  
  
  
# Booking function  
def Booking():  
   
 # used global keyword to  
 # use global variable 'i'  
 global i  
 print(" BOOKING ROOMS")  
 print(" ")  
   
 while 1:  
 n = str(input("Name: "))  
 p1 = str(input("Phone No.: "))  
 a = str(input("Address: "))  
   
 # checks if any field is not empty  
 if n!="" and p1!="" and a!="":  
 name.append(n)  
 add.append(a)  
 break  
   
 else:  
 print("\tName, Phone no. & Address cannot be empty..!!")  
   
 cii=str(input("Check-In: "))  
 checkin.append(cii)  
 cii=cii.split('/')  
 ci=cii  
 ci[0]=int(ci[0])  
 ci[1]=int(ci[1])  
 ci[2]=int(ci[2])  
 date(ci)  
   
 coo=str(input("Check-Out: "))  
 checkout.append(coo)  
 coo=coo.split('/')  
 co=coo  
 co[0]=int(co[0])  
 co[1]=int(co[1])  
 co[2]=int(co[2])  
   
 # checks if check-out date falls after  
 # check-in date  
 if co[1]<ci[1] or co[2]<ci[2]:  
   
 print("\n\tErr..!!\n\tCheck-Out date must fall after Check-In\n")  
 name.pop(i)  
 add.pop(i)  
 checkin.pop(i)  
 checkout.pop(i)  
 Booking()  
 elif co[1]==ci[1] and co[2]>=ci[2] and co[0]<=ci[0]:  
   
 print("\n\tErr..!!\n\tCheck-Out date must fall after Check-In\n")  
 name.pop(i)  
 add.pop(i)  
 checkin.pop(i)  
 checkout.pop(i)  
 Booking()  
 else:  
 pass  
   
 date(co)  
 d1 = datetime.datetime(ci[2],ci[1],ci[0])  
 d2 = datetime.datetime(co[2],co[1],co[0])  
 d = (d2-d1).days  
 day.append(d)  
   
 print("----SELECT ROOM TYPE----")  
 print(" 1. Standard Non-AC")  
 print(" 2. Standard AC")  
 print(" 3. 3-Bed Non-AC")  
 print(" 4. 3-Bed AC")  
 print(("\t\tPress 0 for Room Prices"))  
   
 ch=int(input("->"))  
   
 # if-conditions to display alloted room  
 # type and it's price  
 if ch==0:  
 print(" 1. Standard Non-AC - Rs. 3500")  
 print(" 2. Standard AC - Rs. 4000")  
 print(" 3. 3-Bed Non-AC - Rs. 4500")  
 print(" 4. 3-Bed AC - Rs. 5000")  
 ch=int(input("->"))  
 if ch==1:  
 room.append('Standard Non-AC')  
 print("Room Type- Standard Non-AC")  
 price.append(3500)  
 print("Price- 3500")  
 elif ch==2:  
 room.append('Standard AC')  
 print("Room Type- Standard AC")  
 price.append(4000)  
 print("Price- 4000")  
 elif ch==3:  
 room.append('3-Bed Non-AC')  
 print("Room Type- 3-Bed Non-AC")  
 price.append(4500)  
 print("Price- 4500")  
 elif ch==4:  
 room.append('3-Bed AC')  
 print("Room Type- 3-Bed AC")  
 price.append(5000)  
 print("Price- 5000")  
 else:  
 print(" Wrong choice..!!")  
  
  
 # randomly generating room no. and customer  
 # id for customer  
 rn = random.randrange(40)+300  
 cid = random.randrange(40)+10  
   
   
 # checks if alloted room no. & customer  
 # id already not alloted  
 while rn in roomno or cid in custid:  
 rn = random.randrange(60)+300  
 cid = random.randrange(60)+10  
   
 rc.append(0)  
 p.append(0)  
   
 if p1 not in phno:  
 phno.append(p1)  
 elif p1 in phno:  
 for n in range(0,i):  
 if p1== phno[n]:  
 if p[n]==1:  
 phno.append(p1)  
 elif p1 in phno:  
 for n in range(0,i):  
 if p1== phno[n]:  
 if p[n]==0:  
 print("\tPhone no. already exists and payment yet not done..!!")  
 name.pop(i)  
 add.pop(i)  
 checkin.pop(i)  
 checkout.pop(i)  
 Booking()  
 print("")  
 print("\t\t\t\*\*\*ROOM BOOKED SUCCESSFULLY\*\*\*\n")  
 print("Room No. - ",rn)  
 print("Customer Id - ",cid)  
 roomno.append(rn)  
 custid.append(cid)  
 i=i+1  
 n=int(input("0-BACK\n ->"))  
 if n==0:  
 Home()  
 else:  
 exit()  
  
# ROOMS INFO  
def Rooms\_Info():  
 print(" ------ HOTEL ROOMS INFO ------")  
 print("")  
 print("STANDARD NON-AC")  
 print("---------------------------------------------------------------")  
 print("Room amenities include: 1 Double Bed, Television, Telephone,")  
 print("Double-Door Cupboard, 1 Coffee table with 2 sofa, Balcony and")  
 print("an attached washroom with hot/cold water.\n")  
 print("STANDARD NON-AC")  
 print("---------------------------------------------------------------")  
 print("Room amenities include: 1 Double Bed, Television, Telephone,")  
 print("Double-Door Cupboard, 1 Coffee table with 2 sofa, Balcony and")  
 print("an attached washroom with hot/cold water + Window/Split AC.\n")  
 print("3-Bed NON-AC")  
 print("---------------------------------------------------------------")  
 print("Room amenities include: 1 Double Bed + 1 Single Bed, Television,")  
 print("Telephone, a Triple-Door Cupboard, 1 Coffee table with 2 sofa, 1")  
 print("Side table, Balcony with an Accent table with 2 Chair and an")  
 print("attached washroom with hot/cold water.\n")  
 print("3-Bed AC")  
 print("---------------------------------------------------------------")  
 print("Room amenities include: 1 Double Bed + 1 Single Bed, Television,")  
 print("Telephone, a Triple-Door Cupboard, 1 Coffee table with 2 sofa, ")  
 print("1 Side table, Balcony with an Accent table with 2 Chair and an")  
 print("attached washroom with hot/cold water + Window/Split AC.\n\n")  
 print()  
 n=int(input("0-BACK\n ->"))  
 if n==0:  
 Home()  
 else:  
 exit()  
  
# RESTAURANT FUNCTION  
def restaurant():  
 ph=int(input("Customer Id: "))  
 global i  
 f=0  
 r=0  
 for n in range(0,i):  
 if custid[n]==ph and p[n]==0:  
 f=1  
 print("-------------------------------------------------------------------------")  
 print(" Hotel AnCasa")  
 print("-------------------------------------------------------------------------")  
 print(" Menu Card")  
 print("-------------------------------------------------------------------------")  
 print("\n BEVARAGES 26 Dal Fry................ 140.00")  
 print("---------------------------------- 27 Dal Makhani............ 150.00")  
 print(" 1 Regular Tea............. 20.00 28 Dal Tadka.............. 150.00")  
 print(" 2 Masala Tea.............. 25.00")  
 print(" 3 Coffee.................. 25.00 ROTI")  
 print(" 4 Cold Drink.............. 25.00 ----------------------------------")  
 print(" 5 Bread Butter............ 30.00 29 Plain Roti.............. 15.00")  
 print(" 6 Bread Jam............... 30.00 30 Butter Roti............. 15.00")  
 print(" 7 Veg. Sandwich........... 50.00 31 Tandoori Roti........... 20.00")  
 print(" 8 Veg. Toast Sandwich..... 50.00 32 Butter Naan............. 20.00")  
 print(" 9 Cheese Toast Sandwich... 70.00")  
 print(" 10 Grilled Sandwich........ 70.00 RICE")  
 print(" ----------------------------------")  
 print(" SOUPS 33 Plain Rice.............. 90.00")  
 print("---------------------------------- 34 Jeera Rice.............. 90.00")  
 print(" 11 Tomato Soup............ 110.00 35 Veg Pulao.............. 110.00")  
 print(" 12 Hot & Sour............. 110.00 36 Peas Pulao............. 110.00")  
 print(" 13 Veg. Noodle Soup....... 110.00")  
 print(" 14 Sweet Corn............. 110.00 SOUTH INDIAN")  
 print(" 15 Veg. Munchow........... 110.00 ----------------------------------")  
 print(" 37 Plain Dosa............. 100.00")  
 print(" MAIN COURSE 38 Onion Dosa............. 110.00")  
 print("---------------------------------- 39 Masala Dosa............ 130.00")  
 print(" 16 Shahi Paneer........... 110.00 40 Paneer Dosa............ 130.00")  
 print(" 17 Kadai Paneer........... 110.00 41 Rice Idli.............. 130.00")  
 print(" 18 Handi Paneer........... 120.00 42 Sambhar Vada........... 140.00")  
 print(" 19 Palak Paneer........... 120.00")  
 print(" 20 Chilli Paneer.......... 140.00 ICE CREAM")  
 print(" 21 Matar Mushroom......... 140.00 ----------------------------------")  
 print(" 22 Mix Veg................ 140.00 43 Vanilla................. 60.00")  
 print(" 23 Jeera Aloo............. 140.00 44 Strawberry.............. 60.00")  
 print(" 24 Malai Kofta............ 140.00 45 Pineapple............... 60.00")  
 print(" 25 Aloo Matar............. 140.00 46 Butter Scotch........... 60.00")  
 print("Press 0 -to end ")  
 ch=1  
 while(ch!=0):  
   
 ch=int(input(" -> "))  
   
 # if-elif-conditions to assign item  
 # prices listed in menu card  
 if ch==1 or ch==31 or ch==32:  
 rs=20  
 r=r+rs  
 elif ch<=4 and ch>=2:  
 rs=25  
 r=r+rs  
 elif ch<=6 and ch>=5:  
 rs=30  
 r=r+rs  
 elif ch<=8 and ch>=7:  
 rs=50  
 r=r+rs  
 elif ch<=10 and ch>=9:  
 rs=70  
 r=r+rs  
 elif (ch<=17 and ch>=11) or ch==35 or ch==36 or ch==38:  
 rs=110  
 r=r+rs  
 elif ch<=19 and ch>=18:  
 rs=120  
 r=r+rs  
 elif (ch<=26 and ch>=20) or ch==42:  
 rs=140  
 r=r+rs  
 elif ch<=28 and ch>=27:  
 rs=150  
 r=r+rs  
 elif ch<=30 and ch>=29:  
 rs=15  
 r=r+rs  
 elif ch==33 or ch==34:  
 rs=90  
 r=r+rs  
 elif ch==37:  
 rs=100  
 r=r+rs  
 elif ch<=41 and ch>=39:  
 rs=130  
 r=r+rs  
 elif ch<=46 and ch>=43:  
 rs=60  
 r=r+rs  
 elif ch==0:  
 pass  
 else:  
 print("Wrong Choice..!!")  
 print("Total Bill: ",r)  
   
 # updates restaurant charges and then  
 # appends in 'rc' list  
 r=r+rc.pop(n)  
 rc.append(r)  
 else:  
 pass  
 if f == 0:  
 print("Invalid Customer Id")  
 n=int(input("0-BACK\n ->"))  
 if n==0:  
 Home()  
 else:  
 exit()  
   
   
# PAYMENT FUNCTION   
def Payment():  
   
 ph=str(input("Phone Number: "))  
 global i  
 f=0  
   
 for n in range(0,i):  
 if ph==phno[n] :  
   
 # checks if payment is  
 # not already done  
 if p[n]==0:  
 f=1  
 print(" Payment")  
 print(" --------------------------------")  
 print(" MODE OF PAYMENT")  
   
 print(" 1- Credit/Debit Card")  
 print(" 2- Paytm/PhonePe")  
 print(" 3- Using UPI")  
 print(" 4- Cash")  
 x=int(input("-> "))  
 print("\n Amount: ",(price[n]\*day[n])+rc[n])  
 print("\n Pay For AnCasa")  
 print(" (y/n)")  
 ch=str(input("->"))  
   
 if ch=='y' or ch=='Y':  
 print("\n\n --------------------------------")  
 print(" Hotel AnCasa")  
 print(" --------------------------------")  
 print(" Bill")  
 print(" --------------------------------")  
 print(" Name: ",name[n],"\t\n Phone No.: ",phno[n],"\t\n Address: ",add[n],"\t")  
 print("\n Check-In: ",checkin[n],"\t\n Check-Out: ",checkout[n],"\t")  
 print("\n Room Type: ",room[n],"\t\n Room Charges: ",price[n]\*day[n],"\t")  
 print(" Restaurant Charges: \t",rc[n])  
 print(" --------------------------------")  
 print("\n Total Amount: ",(price[n]\*day[n])+rc[n],"\t")  
 print(" --------------------------------")  
 print(" Thank You")  
 print(" Visit Again :)")  
 print(" --------------------------------\n")  
 p.pop(n)  
 p.insert(n,1)  
   
 # pops room no. and customer id from list and  
 # later assigns zero at same position  
 roomno.pop(n)  
 custid.pop(n)  
 roomno.insert(n,0)  
 custid.insert(n,0)  
   
 else:  
   
 for j in range(n+1,i):  
 if ph==phno[j] :  
 if p[j]==0:  
 pass  
   
 else:  
 f=1  
 print("\n\tPayment has been Made :)\n\n")  
 if f==0:  
 print("Invalid Customer Id")  
   
 n = int(input("0-BACK\n ->"))  
 if n == 0:  
 Home()  
 else:  
 exit()  
  
# RECORD FUNCTION  
def Record():  
   
 # checks if any record exists or not  
 if phno!=[]:  
 print(" \*\*\* HOTEL RECORD \*\*\*\n")  
 print("| Name | Phone No. | Address | Check-In | Check-Out | Room Type | Price |")  
 print("----------------------------------------------------------------------------------------------------------------------")  
   
 for n in range(0,i):  
 print("|",name[n],"\t |",phno[n],"\t|",add[n],"\t|",checkin[n],"\t|",checkout[n],"\t|",room[n],"\t|",price[n])  
   
 print("----------------------------------------------------------------------------------------------------------------------")  
   
 else:  
 print("No Records Found")  
 n = int(input("0-BACK\n ->"))  
 if n == 0:  
 Home()  
   
 else:  
 exit()  
  
# Driver Code  
Home()

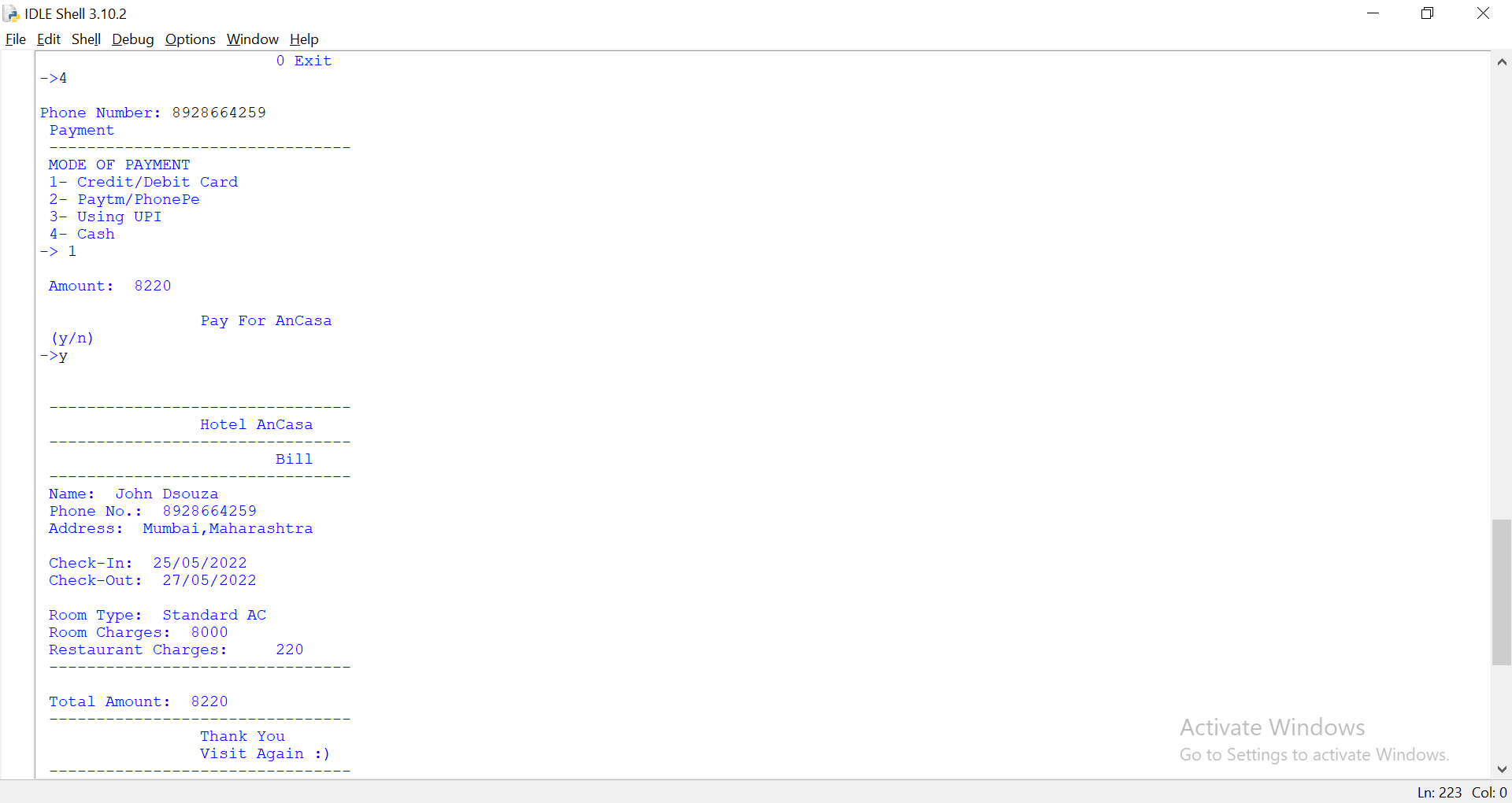
#### **OUTPUT:**



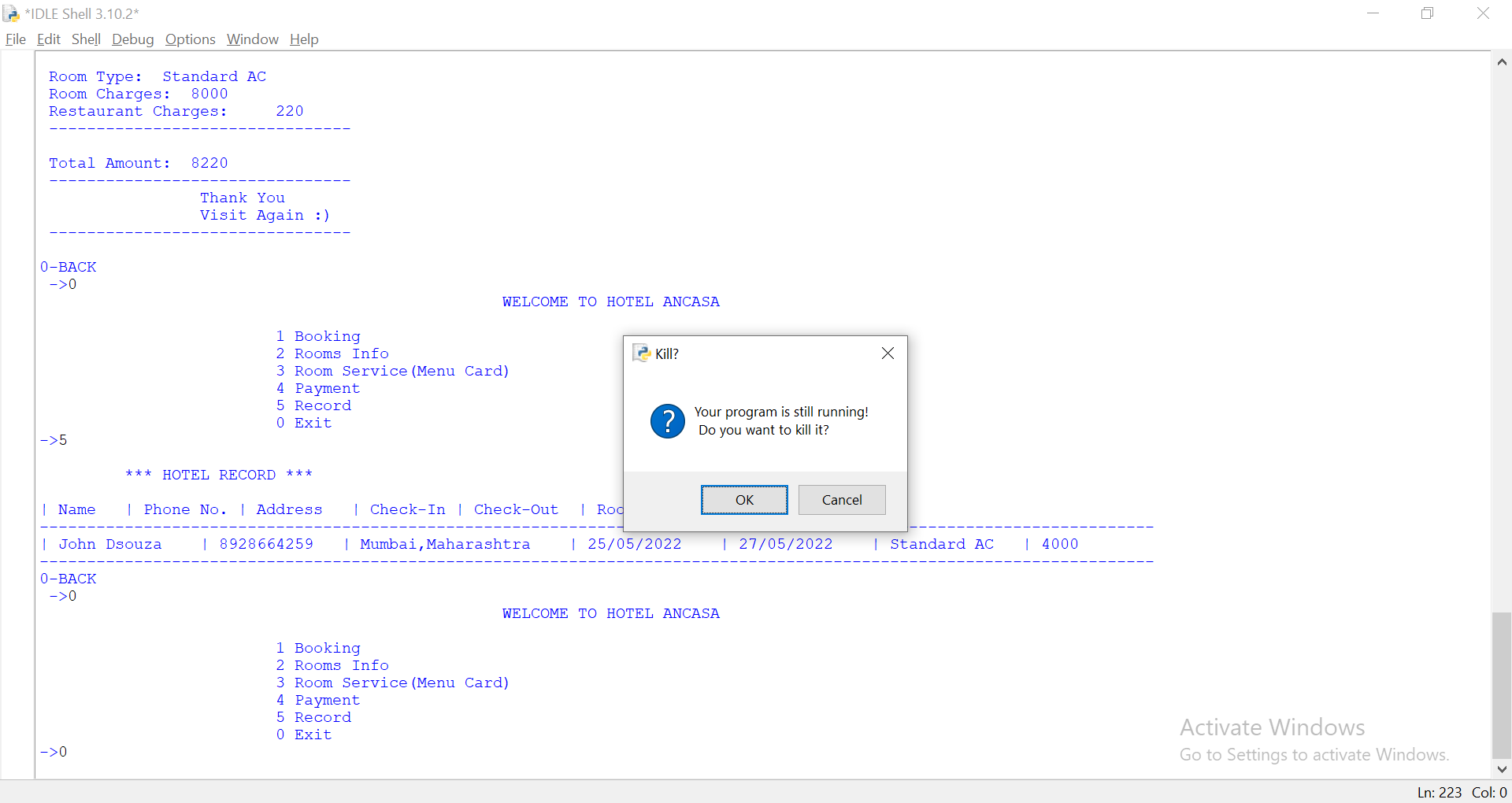












#### **CONCLUSION**

As with all products, the cheapest option is not always the best option – and the same goes for the most expensive choice. There are many inexpensive products that will end up costing you money if you’re not careful when buying a hotel management software system.

Take the time to do some extensive research before investing in the right system for your hotel business. Make sure you get feedback from each of your departments and understand their specific needs before deciding on a solution.

If you do, the benefits can be transformational.

#### **COURSE OUTCOME**

a) Display message on screen using Python script on IDE.

b) Develop python program to demonstrate use of Operators

c) Develop functions for given problem.