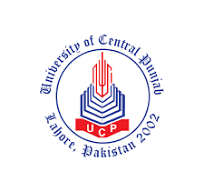
**Introduction to Computing**



**Project Report**

**By**

**Saleha Arshad L1F24BSDS0056**

**Areebah Abbasi L1F24BSDS0073**

**Areeba Aleem L1F24BSDS0082**

**Areeba Faheem L1F24BSDS0092**

**Ayesha Zaman L1F24BSDS0111**

**BS Data Science**

**Course Instructor: Rabia Mehmood**

**Department of Applied Computing Technologies Faculty of Information Technologies (FOIT)**

**Student Portal (Chatbot)**

**Logic**

1. **Introduction:**

This project aims to develop a Student Information Portal that will enable students to access and manage their academic information, including class schedules, assignments, announcements, marks, and quizzes. The system will allow students to view their educational records, assignment deadlines, timetable and classroom venue, etc.

1. **Logic of code:**

* **Student Names:**
  + An array of student names stores the names of 111 students.
  + An array of sections is created and assigned "DA1", "DA2", or "DA3" to each student based on their index.
* **Student Profiles:**
  + - A 2D array of profiles is created to store student profiles.
    - profiles[i][0] stores "ID".
    - profiles[i][1] store the formatted roll number and student name.
    - The code iterates through sections (DA1, DA2, DA3) and assigns each student unique roll numbers and names.
* **Subjects:**
  + An array of subjects stores the names of the courses offered (ITC, BE, IDE, ENG, ITC LAB, BE LAB).
* **Marks:**
  + 2D arrays of midterm marks, quiz marks, and assignment marks store the marks for each student in each subject.
  + The code initializes these arrays with random marks within specified ranges.
* **Reminders and Schedule:**
  + Array reminders and final exam schedule store subject-specific reminders and the final exam schedule.

**2. User Interaction:**

* The code presents a menu to the user with options:
  + Search Profile by Roll Number
  + View Marks
  + View Reminders
  + View Final Exam Schedule
  + Exit
* Based on the user's choice, the code performs the corresponding action:
  + **Search Profile:**
    - Prompts the user to enter a roll number.
    - Validates the roll number.
    - Determines the section and index of the student within the profiles array.
    - Displays the student's profile (section and ID).
  + **View Marks:**
    - Prompts the user to enter a roll number.
    - Validates the roll number.
    - Determines the section and index of the student.
    - Displays the midterm, quiz, and assignment marks for each subject.
  + **View Reminders:**
    - Displays all subject-specific reminders.
  + **View Final Exam Schedule:**
    - Displays the final exam schedule for all subjects.
* The loop continues until the user chooses to exit.

**Concepts used:**

* Arrays: Used to store collections of data (student names, marks, subjects).
* Nested Loops: These are used to iterate through multiple dimensions of the arrays (sections, students, subjects).
* Conditional Statements: These are used to make decisions based on user input (e.g., validating roll number, selecting menu options).
* String Manipulation: Used to format and display student IDs and names.

This code shows a basic student information portal with features like profile search, mark viewing, and important information display.

**Pseudo-code:**

**1. Initialise System:**

• Print welcome messages for the chatbot.

**2. Define Data:**

• Create a list of student names of 111 predefined student names.

• Assign sections (DA1, DA2, DA3) based on roll numbers:

• Roll numbers 1–37 → DA1

• Roll numbers 38–74 → DA2

• Roll numbers 75–111 → DA3

• Create a 2D list of profiles to store each student’s profile (ID and name).

• Define subjects: ITC, BE, IDE, ENG, ITC LAB, BE LAB.

• Initialize arrays for:

• Midterm marks

• Quiz marks

• Assignment marks

Each is initialized with specific ranges.

**3. Setup Reminders and Schedules:**

• Define reminders for each subject and section.

• Define the final exam schedule for each subject and section.

• Define the class schedule for each subject and section.

**4. Main Menu:**

• Display a greeting and main menu options:

* Search Profile by Roll Number
* View Marks
* View Reminders
* View Final Exam Schedule
* View Class Schedule
* Exit

**5. Menu Options:**

• Option 1: Search Profile by Roll Number

• Prompt for roll number.

• Validate the input (1–111).

• Display the student’s profile (ID, name, section).

• Option 2: View Marks

• Prompt for roll number.

• Validate the input.

• Display the marks for all subjects (midterms, quizzes, assignments).

• Option 3: View Reminders

• Prompt for roll number.

• Determine the section (DA1, DA2, DA3).

• Display reminders specific to the section.

• Option 4: View Final Exam Schedule

• Prompt for roll number.

• Determine the section.

• Display the final exam schedule for the section.

• Option 5: View Class Schedule

• Prompt for roll number.

• Determine the section.

• Display the class schedule for the section.

• Option 6: Exit

• Display a goodbye message and terminate the program.

**6. Loop Until Exit:**

• Keep displaying the main menu until the user chooses the exit option.

**Code:-**

#include <iostream>

#include <string>

using namespace std;

int main() {

// Define student names

cout << "UNIVERSITY OF CENTRAL PUNAJB" << endl;

cout << "This chatbot is only for data science semester 1 students" << endl;

string studentNames[111] = {

"Ahmed", "Ali", "Fatima", "Ayesha", "Hassan", "Hussain", "Zainab", "Saad", "Amna", "Maryam",

"Yusuf", "Ibrahim", "Musa", "Nuh", "Ismail", "Eesa", "Sharique","Muzamil","Areebah","Areeba",

"Khadija", "Sumayya", "Bilal", "Hamza", "Umar", "Uthman", "Abu Bakr", "Abdul Rahman", "Abdul Aziz", "Abdul Qadir",

"Zain", "Rayan", "Anas", "Sara", "Huda", "Noor", "Tariq", "Imran", "Jawad", "Asim",

"Usman", "Rafia", "Samira", "Shahid", "Sadia", "Rashid", "Nadia", "Ishaq", "Jafar", "Furqan",

"Naeem", "Farooq", "Shaheen", "Zubair", "Anwar", "Sameer", "Lubna", "Shamim", "Ibtisam", "Asad",

"Zahra", "Hiba", "Rabia", "Adeel", "Fahad", "Faizan", "Hafsa", "Shazia", "Aqsa", "Basit",

"Qasim", "Mahmood", "Tahira", "Arsalan", "Sana", "Areeb", "Madiha", "Azhar", "Saqib", "Uzair",

"Inaya", "Bushra", "Faisal", "Kashif", "Haroon", "Ubaid", "Salma", "Talha", "Yumna", "Aiman",

"Hira", "Eman", "Aqeel", "Junaid", "Tuba", "Laiba", "Rania", "Maham", "Sadia", "Mehwish",

"Zara", "Kiran", "Nida"

};

// Assign sections DA1, DA2, and DA3 to students

string sections[111];

for (int i = 0; i < 111; i++) {

if (i < 37) {

sections[i] = "DA1";

}

else if (i < 74) {

sections[i] = "DA2";

}

else {

sections[i] = "DA3";

}

}

string profiles[111][2];

for (int i = 0; i < 111; i++) {

profiles[i][0] = "ID";

profiles[i][1] = (i + 1 < 10 ? "00" : (i + 1 < 100 ? "0" : "")) + to\_string(i + 1) + " - " + studentNames[i];

}

// Subjects

string subjects[6] = { "ITC", "BE", "IDE", "ENG", "ITC LAB", "BE LAB" };

// Marks for midterms, quizzes, and assignments by subject for each student

int midtermMarks[111][6];

int quizMarks[111][6];

int assignmentMarks[111][6];

// Initialize marks with deterministic values

for (int student = 0; student < 111; student++) {

for (int subject = 0; subject < 6; subject++) {

midtermMarks[student][subject] = 75 + (student + subject) % 26; // Midterm marks: 75-100

quizMarks[student][subject] = 10 + (student + subject \* 2) % 16; // Quiz marks: 10-25

assignmentMarks[student][subject] = 15 + (student + subject \* 3) % 16; // Assignment marks: 15-30

}

}

// Reminders for each subject, divided by section

string reminders[6][3] = {

{

"ITC: Complete project by 25th January for DA1",

"ITC: Complete project by 25th January for DA2",

"ITC: Complete project by 25th January for DA3"

},

{

"BE: Submit lab report by 26th January for DA1",

"BE: Submit lab report by 26th January for DA2",

"BE: Submit lab report by 26th January for DA3"

},

{

"IDE: Finalize design draft by 27th January for DA1",

"IDE: Finalize design draft by 27th January for DA2",

"IDE: Finalize design draft by 27th January for DA3"

},

{

"ENG: Prepare for presentation by 28th January for DA1",

"ENG: Prepare for presentation by 28th January for DA2",

"ENG: Prepare for presentation by 28th January for DA3"

},

{

"ITC LAB: Complete experiment by 29th January for DA1",

"ITC LAB: Complete experiment by 29th January for DA2",

"ITC LAB: Complete experiment by 29th January for DA3"

},

{

"BE LAB: Finish practical work by 30th January for DA1",

"BE LAB: Finish practical work by 30th January for DA2",

"BE LAB: Finish practical work by 30th January for DA3"

}

};

// Final exam schedule, divided by section

string finalExamSchedule[6][3] = {

{

"ITC: 10:00 AM, 1st February for DA1",

"ITC: 10:00 AM, 1st February for DA2",

"ITC: 10:00 AM, 1st February for DA3"

},

{

"BE: 12:00 PM, 2nd February for DA1",

"BE: 12:00 PM, 2nd February for DA2",

"BE: 12:00 PM, 2nd February for DA3"

},

{

"ITC LAB: 9:00 AM, 3rd February for DA1",

"ITC LAB: 9:00 AM, 3rd February for DA2",

"ITC LAB: 9:00 AM, 3rd February for DA3"

},

{

"BE LAB: 11:00 AM, 4th February for DA1",

"BE LAB: 11:00 AM, 4th February for DA2",

"BE LAB: 11:00 AM, 4th February for DA3"

},

{

"IDE: 1:00 PM, 5th February for DA1",

"IDE: 1:00 PM, 5th February for DA2",

"IDE: 1:00 PM, 5th February for DA3"

},

{

"ENG: 10:00 AM, 6th February for DA1",

"ENG: 10:00 AM, 6th February for DA2",

"ENG: 10:00 AM, 6th February for DA3"

}

};

// Class schedule, divided by section

string classSchedule[6][3] = {

{

"ITC: Monday & Wednesday, 9:00 AM - 10:30 AM for DA1",

"ITC: Monday & Wednesday, 9:00 AM - 10:30 AM for DA2",

"ITC: Monday & Wednesday, 9:00 AM - 10:30 AM for DA3"

},

{

"BE: Tuesday & Thursday, 10:45 AM - 12:15 PM for DA1",

"BE: Tuesday & Thursday, 10:45 AM - 12:15 PM for DA2",

"BE: Tuesday & Thursday, 10:45 AM - 12:15 PM for DA3"

},

{

"IDE: Monday & Wednesday, 12:30 PM - 2:00 PM for DA1",

"IDE: Monday & Wednesday, 12:30 PM - 2:00 PM for DA2",

"IDE: Monday & Wednesday, 12:30 PM - 2:00 PM for DA3"

},

{

"ENG: Tuesday & Thursday, 2:15 PM - 3:45 PM for DA1",

"ENG: Tuesday & Thursday, 2:15 PM - 3:45 PM for DA2",

"ENG: Tuesday & Thursday, 2:15 PM - 3:45 PM for DA3"

},

{

"ITC LAB: Friday, 9:00 AM - 11:00 AM for DA1",

"ITC LAB: Friday, 9:00 AM - 11:00 AM for DA2",

"ITC LAB: Friday, 9:00 AM - 11:00 AM for DA3"

},

{

"BE LAB: Friday, 11:15 AM - 1:15 PM for DA1",

"BE LAB: Friday, 11:15 AM - 1:15 PM for DA2",

"BE LAB: Friday, 11:15 AM - 1:15 PM for DA3"

}

};

cout << "Welcome to the Student Information Portal for DATA SCIENCE SEM 1! I'm here to help you explore student profiles, marks, and more.\n";

cout << "Before we start, how's your day going? I hope it's great! Let's dive in.\n";

while (true) {

cout << "\nMain Menu\n";

cout << "1. Search Profile by Roll Number\n";

cout << "2. View Marks\n";

cout << "3. View Reminders\n";

cout << "4. View Final Exam Schedule\n";

cout << "5. View Class Schedule\n";

cout << "6. Exit\n";

int choice;

cout << "Enter your choice (1-6): ";

cin >> choice;

if (choice == 1) {

int rollNumber;

cout << "Enter Roll Number (1-111): ";

cin >> rollNumber;

if (rollNumber >= 1 && rollNumber <= 111) {

cout << "\nGreat choice! Let’s look at the profile for roll number " << rollNumber << ".\n";

// Display Student Profile

cout << "Student Profile:\n";

cout << profiles[rollNumber - 1][0] << ": " << profiles[rollNumber - 1][1] << "\n";

cout << "Section: " << sections[rollNumber - 1] << "\n";

cout << "Did you find the information helpful?\n";

}

else {

cout << "Invalid Roll Number. Please enter a number between 1 and 111.\n";

}

}

else if (choice == 2) {

int rollNumber;

cout << "Enter Roll Number (1-111): ";

cin >> rollNumber;

if (rollNumber >= 1 && rollNumber <= 111) {

cout << "\nLet's review the marks for roll number " << rollNumber << ".\n";

// Display Marks

cout << "Marks:\n";

for (int i = 0; i < 6; i++) {

cout << subjects[i] << ":\n";

cout << " Midterm Marks: " << midtermMarks[rollNumber - 1][i] << "\n";

cout << " Quiz Marks: " << quizMarks[rollNumber - 1][i] << "\n";

cout << " Assignment Marks: " << assignmentMarks[rollNumber - 1][i] << "\n";

}

cout << "Did any of these scores surprise you?\n";

}

else {

cout << "Invalid Roll Number. Please enter a number between 1 and 111.\n";

}

}

else if (choice == 3) {

int rollNumber;

cout << "Enter Roll Number (1-111): ";

cin >> rollNumber;

if (rollNumber >= 1 && rollNumber <= 111) {

string section = sections[rollNumber - 1];

int sectionIndex = (section == "DA1" ? 0 : (section == "DA2" ? 1 : 2));

// Display Reminders

cout << "\nHere are some reminders to keep you on track for section " << section << ":\n";

for (int i = 0; i < 6; i++) {

cout << "- " << reminders[i][sectionIndex] << "\n";

}

cout << "Which reminder do you think is the most important for you right now?\n";

}

else {

cout << "Invalid Roll Number. Please enter a number between 1 and 111.\n";

}

}

else if (choice == 4) {

int rollNumber;

cout << "Enter Roll Number (1-111): ";

cin >> rollNumber;

if (rollNumber >= 1 && rollNumber <= 111) {

string section = sections[rollNumber - 1];

int sectionIndex = (section == "DA1" ? 0 : (section == "DA2" ? 1 : 2));

// Display Final Exam Schedule

cout << "\nFinal Exam Schedule for section " << section << ":\n";

for (int i = 0; i < 6; i++) {

cout << "- " << finalExamSchedule[i][sectionIndex] << "\n";

}

cout << "Are you prepared for these exams? Which subject are you most confident about?\n";

}

else {

cout << "Invalid Roll Number. Please enter a number between 1 and 111.\n";

}

}

else if (choice == 5) {

int rollNumber;

cout << "Enter Roll Number (1-111): ";

cin >> rollNumber;

if (rollNumber >= 1 && rollNumber <= 111) {

string section = sections[rollNumber - 1];

int sectionIndex = (section == "DA1" ? 0 : (section == "DA2" ? 1 : 2));

// Display Class Schedule

cout << "\nClass Schedule for section " << section << ":\n";

for (int i = 0; i < 6; i++) {

cout << "- " << classSchedule[i][sectionIndex] << "\n";

}

cout << "Which class are you most excited about attending?\n";

}

else {

cout << "Invalid Roll Number. Please enter a number between 1 and 111.\n";

}

}

else if (choice == 6) {

cout << "It was great assisting you! Have a wonderful day ahead. Goodbye!\n";

break;

}

else {

cout << "Invalid choice. Let’s try again!\n";

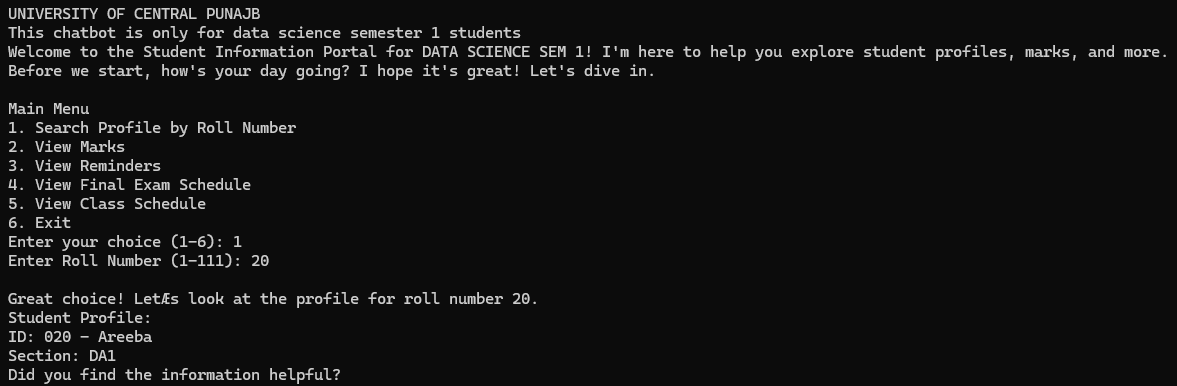
}

}

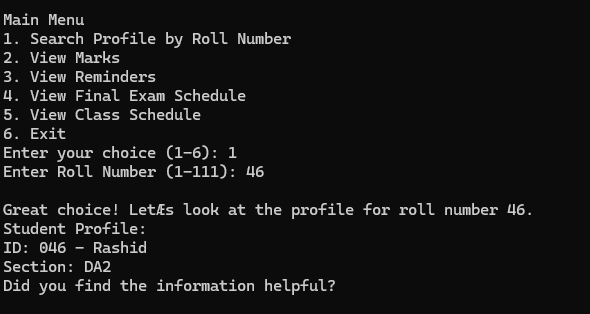
return 0;

}

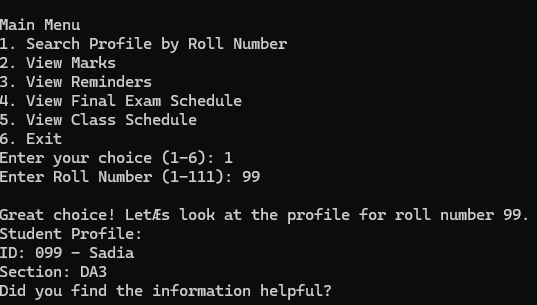
**Output 1:**



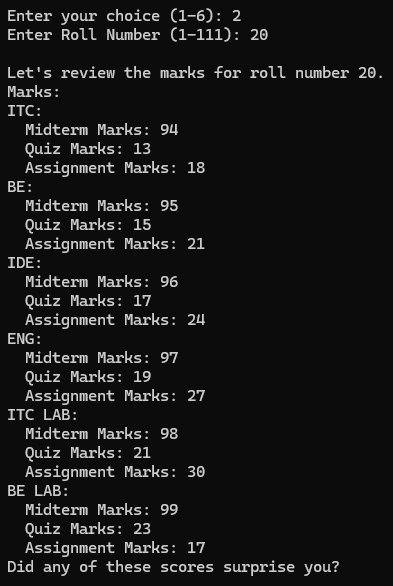
**Output 2:**

****

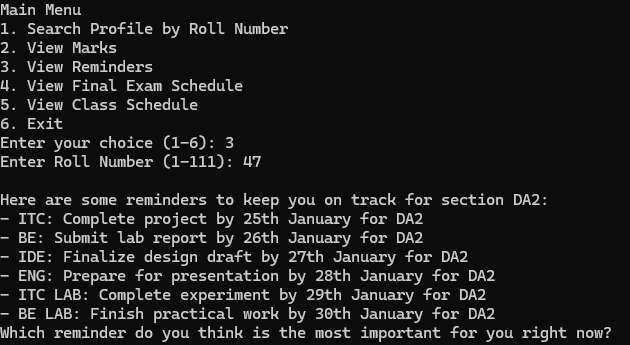
**Output 3:**

****

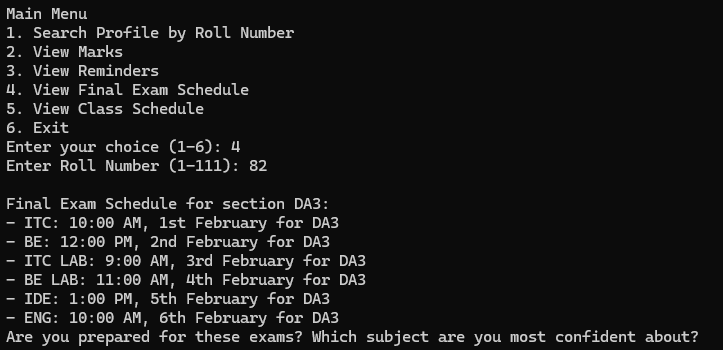
**Output (View Marks):**



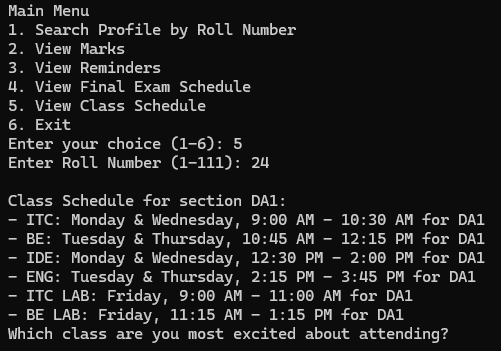
**Output (View Reminders):**

****

**Output (View Final Exam Schedule):**

****

**Output (View Class Schedule):**

****

**Flow Chart:**

**“Student Information Portal”**

