A

Project Report

on

Student Management System

Submitted to

Chitkara University, Punjab

for

B.tech program

by

Name: Tanvi Gupta

Roll No: 2210990905

Under the supervision of :

Miss Shilpi Garg

Associate professor

Department of computer applications

M

INDEX

ACKNOWLEDGEMENT

It is an honor for me to thank all those people who made this thesis possible. I want to thank my advisor. This would not be possible without their help and support.

I also want to thank all my faculties and friends for their constant help throughout the whole time. I could not have done it without their help. I would also want to thank my parents for helping me to complete this work in every single step.

INTRODUCTION

Schools and Universities are the foundation of knowledge and an educational body on which students rely upon. Therefore, they need to maintain a proper database of its students to keep all the updated records and easily share information with students.

Most schools and Universities count on an advanced software tool knows as 'Student Information System (SIS)' to keep all their student records and administrative operations including, examinations, attendance, and other activities.

Over the recent years, the performance and efficiency of the education industry have been enhanced by using the Student Management System. This tool has productively taken over the workload of the admin department with its well-organized, easy, and reliable online school management software.

ABSTRACT

In this software we can register as a user and user has two types :

- Administrator
- Student

Administrator has a power to add new user and can edit and delete a user. A student can register as a user and can edit and delete his profile. The administrator can edit and delete the marks of the student.

All the users can see the marks. A student management system deals with the various activities related to the students

APPLICATION AND ADVANTAGES

- It helps you manage all the student related data in a well-organized manner.
- It improves overall productivity of universities or colleges.
- Keeps a track of student’s grades and also stores personal information.
- Some applications are Progress tracking and lesson management, Secure messaging , School management, Attendance management, Student admissions and application process

CONCLUSION

Our project is only a humble venture to satisfy the needs in an Institution. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses. Last but not least it is no the work that played the ways to success but ALMIGHTY.

FUTURE SCOPE

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project :

1. Discontinue of particular student eliminate potential attendance.

2.Bar code Reader based attendance system.

3.Individual attendance system with photo using student login.

SOURCE CODE

```
import pickle
import time
import os

def set_data():
    print("ENTER STUDENT'S DETAILS")
    rollno = int(input('Enter roll number: '))
    name = input('Enter name: ')
    english = int(input('Enter Marks in English: '))
    maths = int(input('Enter Marks in Maths: '))
    physics = int(input('Enter Marks in Physics: '))
    chemistry = int(input('Enter Marks in Chemistry: '))
    cs = int(input('Enter Marks in CS: '))
    print()

    #create a dictionary
    student = {}
    student['rollno'] = rollno
    student['name'] = name
    student['english'] = english
    student['maths'] = maths
    student['physics'] = physics
    student['chemistry'] = chemistry
    student['cs'] = cs
    return student


def display_data(student):
    print('\nSTUDENT DETAILS..')
    print('Roll Number:', student['rollno'])
    print('Name:', student['name'])
    print('English:', student['english'])
    print('Maths:', student['maths'])
    print('Physics:', student['physics'])
    print('Chemistry:', student['chemistry'])
    print('CS:', student['cs'])

def display_data_tabular(student):
    print('{0:<8}{1:<20}{2:<10}{3:<10}{4:<10}{5:<10}{6:<10}'.format(stu
dent['rollno'],
        student['name'], student['english'],student['maths'],
student['physics'],
        student['chemistry'],student['cs']))

def class_result():
    #open file in binary mode for reading
    try:
        infile = open('student.dat', 'rb')
    except FileNotFoundError:
        print('No record found..')
        print('Go to admin menu to create record')
        return

    print('{0:<8}{1:<20}{2:<10}{3:<10}{4:<10}{5:<10}{6:<10}'.format('Ro
llno', 'Name', 'English', 'Maths','Physics','Chemistry','CS'))
    #read to the end of file.
    while True:
        try:
```

```
#reading the oject from file
            student = pickle.load(infile)

            #display the record
            display_data_tabular(student)
        except EOFError:
            break

    #close the file
    infile.close()

def write_record():
    #open file in binary mode for writing.
    outfile = open('student.dat', 'ab')

    while(True):
        #serialize the record and writing to file
        pickle.dump(set_data(), outfile)
        ans = input('Wants to enter more record (y/n)?: ')
        if ans in 'nN':
            break

    #close the file
    outfile.close()


def read_records():
    #open file in binary mode for reading
    try:
        infile = open('student.dat', 'rb')
    except FileNotFoundError:
        print('No record found..')
        return

    #read to the end of file.
    while True:
        try:
            #reading the oject from file
            student = pickle.load(infile)

            #display the record
            display_data(student)
        except EOFError:
            break

    #close the file
    infile.close()

def search_record():
    #open file in binary mode for reading
    try:
        infile = open('student.dat', 'rb')
    except FileNotFoundError:
        print('No record..')
        return

    found = False
    print('SEARCH RECORD')
print('\nMODIFY RECORD')
    try:
        infile = open('student.dat', 'rb')
    except FileNotFoundError:
        print('No record found to modify..')
        return

    found = False
    outfile = open("temp.dat","wb")
    rollno = int(input('Enter roll number: '))
    while True:
        try:
            #reading the oject from file
            student = pickle.load(infile)

            #display record if found and set flag
            if student['rollno'] == rollno:

                print('Name:',student['name'])
                ans=input('Wants to edit(y/n)? ')
                if ans in 'yY':
                    student['name'] = input("Enter the name ")

                print('English marks:',student['english'])
                ans=input('Wants to edit(y/n)? ')
                if ans in 'yY':
                    student['english'] = int(input("Enter new marks: 
"))

                print('Maths marks:',student['maths'])
                ans=input('Wants to edit(y/n)? ')
                if ans in 'yY':
                    student['maths'] = int(input("Enter new marks: "))

                print('Physics marks:',student['physics'])
                ans=input('Wants to edit(y/n)? ')
                if ans in 'yY':
                    student['physics'] = int(input("Enter new marks: 
"))

                print('Chemistry marks:',student['chemistry'])
                ans=input('Wants to edit(y/n)? ')
                if ans in 'yY':
                    student['chemistry'] = int(input("Enter new marks: 
"))

                print('CS marks:',student['cs'])
                ans=input('Wants to edit(y/n)? ')
                if ans in 'yY':
                    student['cs'] = int(input("Enter new marks: "))

                pickle.dump(student,outfile)
                found = True
                break
            else:
                pickle.dump(student,outfile)
        except EOFError:
            break
    if found == False:
        print('Record not Found')
    else:
        print('Record updated')
        display_data(student)

    infile.close()
    outfile.close()
    os.remove("student.dat")
    os.rename("temp.dat","student.dat")

def intro():
    print("="*80)
    print("{: ^80s}".format("STUDENT"))
    print("{: ^80s}".format("REPORT CARD"))
    print("{: ^80s}".format("PROJECT"))
    print("{: ^80s}".format("MADE BY: PyForSchool.com"))
    print("="*80)
    print()

def main_menu():
    time.sleep(1)
    print("MAIN MENU")
    print("1. REPORT MENU")
    print("2. ADMIN MENU")
    print("3. EXIT")

def report_menu():
    time.sleep(1)
    print("REPORT MENU")
    print("1. CLASS RESULT")
    print("2. STUDENT REPORT CARD")
    print("3. BACK TO MAIN MENU")

def admin_menu():
    time.sleep(1)
    print("\nADMIN MENU")
    print("1. CREATE STUDENT RECORD")
    print("2. DISPLAY ALL STUDENTS RECORDS")
    print("3. SEARCH STUDENT RECORD ")
    print("4. MODIFY STUDENT RECORD ")
    print("5. DELETE STUDENT RECORD ")
    print("6. BACK TO MAIN MENU")

def main():
    intro()
    while(True):
        main_menu()
        choice = input('Enter choice(1-3): ')
        print()

        if choice == '1':
            while True:
                report_menu()
                rchoice = input('Enter choice(1-3): ')
                print()
                if rchoice == '1':
                    class_result()
                elif rchoice == '2':
                    search_record()
                elif rchoice == '3':
                    break
                else:
                    print('Invalid input !!!\n')
                print()

        elif choice == '2':
            while True:
                admin_menu()
                echoice = input('Enter choice(1-6): ')
                print()
                if echoice == '1':
                    write_record()
                elif echoice == '2':
                    read_records()
                elif echoice == '3':
                    search_record()
                elif echoice == '4':
                    modify_record()
                elif echoice == '5':
                    delete_record()
                elif echoice == '6':
                    break
                else:
                    print('Invalid input !!!\n')


        elif choice == '3':
            print('Thanks for using Student Management System')
            break
        else:
            print('Invalid input!!!')
            print()

#call the main function.
main()
```

OUTPUT

```
STUDENT
                                  REPORT CARD
                                    PROJECT
                            MADE BY: PyForSchool.com
=======================================================================
=========

MAIN MENU
1. REPORT MENU
2. ADMIN MENU
3. EXIT
Enter choice(1-3): 2


ADMIN MENU
1. CREATE STUDENT RECORD
2. DISPLAY ALL STUDENTS RECORDS
3. SEARCH STUDENT RECORD
4. MODIFY STUDENT RECORD
5. DELETE STUDENT RECORD
6. BACK TO MAIN MENU
Enter choice(1-6): 1

ENTER STUDENT'S DETAILS
Enter roll number: 05
Enter name: tanvi
Enter Marks in English: 95
Enter Marks in Maths: 93
Enter Marks in Physics: 65
Enter Marks in Chemistry: 75
Enter Marks in CS: 76

Wants to enter more record (y/n)?: n

ADMIN MENU
1. CREATE STUDENT RECORD
2. DISPLAY ALL STUDENTS RECORDS
3. SEARCH STUDENT RECORD
4. MODIFY STUDENT RECORD
5. DELETE STUDENT RECORD
6. BACK TO MAIN MENU
Enter choice(1-6): 3

SEARCH RECORD
Enter the rollno you want to search: 905
Record not found!!

ADMIN MENU
1. CREATE STUDENT RECORD
2. DISPLAY ALL STUDENTS RECORDS
3. SEARCH STUDENT RECORD
4. MODIFY STUDENT RECORD
5. DELETE STUDENT RECORD
6. BACK TO MAIN MENU
Enter choice(1-6): 3

SEARCH RECORD
Enter the rollno you want to search: 905
```

```
Record not found!!

ADMIN MENU
1. CREATE STUDENT RECORD
2. DISPLAY ALL STUDENTS RECORDS
3. SEARCH STUDENT RECORD
4. MODIFY STUDENT RECORD
5. DELETE STUDENT RECORD
6. BACK TO MAIN MENU
Enter choice(1-6): 3

SEARCH RECORD
Enter the rollno you want to search: 05

STUDENT DETAILS..
Roll Number: 5
Name: tanvi
English: 95
Maths: 93
Physics: 65
Chemistry: 75
CS: 76

ADMIN MENU
1. CREATE STUDENT RECORD
2. DISPLAY ALL STUDENTS RECORDS
3. SEARCH STUDENT RECORD
4. MODIFY STUDENT RECORD
5. DELETE STUDENT RECORD
6. BACK TO MAIN MENU
Enter choice(1-6): 4


MODIFY RECORD
Enter roll number: 05
Name: tanvi
Wants to edit(y/n)? n
English marks: 95
Wants to edit(y/n)? n
Maths marks: 93
Wants to edit(y/n)? n
Physics marks: 65
Wants to edit(y/n)? n
Chemistry marks: 75
Wants to edit(y/n)? 98
CS marks: 76
Wants to edit(y/n)? n
Record updated

STUDENT DETAILS..
Roll Number: 5
Name: tanvi
English: 95
Maths: 93
Physics: 65
Chemistry: 75
CS: 76
```

```
ADMIN MENU
1. CREATE STUDENT RECORD
2. DISPLAY ALL STUDENTS RECORDS
3. SEARCH STUDENT RECORD
4. MODIFY STUDENT RECORD
5. DELETE STUDENT RECORD
6. BACK TO MAIN MENU
Enter choice(1-6): 6

MAIN MENU
1. REPORT MENU
2. ADMIN MENU
3. EXIT
Enter choice(1-3): 3

Thanks for using Student Management System
```

PROJECT TEMPLATE

<table>
<tr><td>Group Name</td><td><b>Group 29 B</b></td></tr>
<tr><td>Project Title</td><td>Student management system</td></tr>
<tr><td>Team Leader</td><td>Tanvi gupta</td></tr>
<tr><td>Details of Work division</td><td>
<table>
<tr><th>Student/rollno</th><th>Work / Role Allotted</th></tr>
<tr><td></td><td></td></tr>
<tr><td>Tanvi / 2210990905</td><td>presentation</td></tr>
<tr><td>Teena / 2210990910</td><td>coding</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>High level Approach to be followed:
● Share the details in points</td><td>Student management project manages all the information about the students.
It can also modify the information.</td></tr>
<tr><td>Lower level Approach to be followed:
● Share the details in points</td><td>There are two sections:
1. Admin section
2. Student section
Admin section can insert, delete, update data and database.
Student section can search their record and marksheet here.</td></tr>
<tr><td>Advantage of the project with applicability:
Highlight the unique feature(s) of the project if any</td><td>1. it helps you manage all the student related data in a well-organized manner.
2. improves overall productivity of universities/colleges.
3. keep a track of student’s grades, also stores personal information.</td></tr>
<tr><td>Schedule for implementing the use case:
● Share the Tentative Date of Completion of Expected Deliverables</td><td>
<table>
<tr><th>Deliverable</th><th>Expected Date of Completion</th></tr>
<tr><td>User case 1</td><td>8 dec,2022</td></tr>
<tr><td>User case 2</td><td>17 dec,2022</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Future scope of the project
● Share the details in points</td><td>1.Helps in making the process of managing information easier and to also make it accessible.
2.Helps to coordinate scheduling and communications between faculty regarding students.</td></tr>
</table>

DOCUMENT HISTORY:

Created By	Tanvi Gupta , Teena Goyal
Approved By	Miss Shilpi Garg
Month of Creation	December