



Mid-Term Project Report

Course-Name: Programming in Python

Section: A

Submitted to: DR. ABDUS SALAM

Submitted By: Shahriar Soudeep

Group Members:

Name	Id
Shahriar Soudeep	20-43823-2
Most. Lilun Nahar Aurthy	20-43997-2

Project-Title: Wedding Event Management System.

Project Description:

The Wedding Event Management System is a command-line application designed from scratch to assist users in managing wedding event details, including the bride's and groom's names, wedding date, venue, and budgeting information. This project provides various features, including adding, displaying, updating, and deleting marriage details. Additionally, it allows users to calculate the remaining budget for each wedding event. This report will describe the key features of the project and their implementation details.

Implemented Features:

1. Adding Marriage Details

The “Add Marriage Details” feature allows users to input information about a wedding event, such as the groom's name, bride's name, wedding date, venue, total budget, catering cost, and shopping cost. The system calculates the total expense by adding the catering and shopping costs. In case of invalid cost inputs, an error message is displayed because here implement the exception handling techniques. Here is the relevant **code segment**:

def add_marriage_details():

```
    groom_name = input("Enter Groom's Name: ")
    bride_name = input("Enter Bride's Name: ")
    date = input("Enter Wedding Date: ")
    venue = input("Enter Venue: ")
    try:
        total_budget = float(input("Enter Total Budget: "))
        catering_cost = float(input("Enter Catering Cost: "))
        shopping_cost = float(input("Enter Shopping Cost: "))
        total_expense = catering_cost + shopping_cost
    except ValueError:
        print("Invalid cost input. Please enter a valid number.")
        return
    wedding = {
        'Groom Name': groom_name,
        'Bride Name': bride_name,
        'Date': date,
```

```

        'Venue': venue,
        'Total Budget': total_budget,
        'Total Expense': total_expense
    }
    wedding_data.append(wedding)
    print("Marriage details added successfully!")

```

Output:

```

Wedding Event Management System
1. Add Marriage Details
2. Show Details
3. Update Details
4. Delete Details
5. Calculate Remaining Amount
6. Exit
Enter your choice: 1
Enter Groom's Name: Biplob
Enter Bride's Name: Aurthy
Enter Wedding Date: 10-21-2023
Enter Venue: Rangpur
Enter Total Budget: 200000
Enter Catering Cost: 50000
Enter Shopping Cost: 80000
Marriage details added successfully!

```

2. Showing Marriage Details

The “Show Details” feature allows users to display the details of all recorded wedding events which user give into “Add Marriage Details” feature section. It iterates through the **wedding_data** list and prints out the details of each wedding, including groom's name, bride's name, date, venue, total budget, and total expense. Here is the relevant **code segment**:

def show_details():

```

for index, wedding in enumerate(wedding_data, 1):
    print(f"Marriage {index} details:")
    for key, value in wedding.items():
        print(f"{key}: {value}")

```

```
print()
```

Output:

```
Wedding Event Management System
1. Add Marriage Details
2. Show Details
3. Update Details
4. Delete Details
5. Calculate Remaining Amount
6. Exit
Enter your choice: 2
Marriage 1 details:
Groom Name: Biplob
Bride Name: Aurthy
Date: 10-21-2023
Venue: Rangpur
Total Budget: 200000.0
Total Expense: 130000.0
```

3. Updating Marriage Details

The “**Update Details**” feature enables users to update wedding event information based on the groom's name. Users can modify the venue and total budget for a specific wedding event. If the groom's name is not found, a message indicating that the marriage details were not found is displayed. Here is the relevant **code segment**:

def update_details():

```
groom_name = input("Enter Groom's Name to update details: ")
for wedding in wedding_data:
    if wedding['Groom Name'] == groom_name:
        venue = input("Enter New Venue: ")
        total_budget = float(input("Enter New Total Budget: "))
        wedding['Venue'] = venue
        wedding['Total Budget'] = total_budget
        print("Details updated successfully!")
    return
```

```
print("Marriage details not found.")
```

Output:

```
Wedding Event Management System
1. Add Marriage Details
2. Show Details
3. Update Details
4. Delete Details
5. Calculate Remaining Amount
6. Exit
Enter your choice: 3
Enter Groom's Name to update details: Soudeep
Marriage details not found.

Wedding Event Management System
1. Add Marriage Details
2. Show Details
3. Update Details
4. Delete Details
5. Calculate Remaining Amount
6. Exit
Enter your choice: Biplob
Invalid choice. Please try again.

Wedding Event Management System
1. Add Marriage Details
2. Show Details
3. Update Details
4. Delete Details
5. Calculate Remaining Amount
6. Exit
Enter your choice: 3
Enter Groom's Name to update details: Biplob
Enter New Venue: Dhaka
Enter New Total Budget: 300000
Details updated successfully!
```

4. Deleting Marriage Details

The “Delete Details” feature allows users to remove a wedding event's details based on the bride's name. If the bride's name is found, the details are removed from the **wedding_data** list. If not found, a message indicating that the marriage details were not found is displayed. Here is the relevant **code segment**:

```
def delete_details():
```

```
    bride_name = input("Enter Bride's Name to delete details: ")
```

```

for wedding in wedding_data:
    if wedding['Bride Name'] == bride_name:
        wedding_data.remove(wedding)
        print("Details deleted successfully!")
        return
print("Marriage details not found.")

```

Output:

```

Wedding Event Management System
1. Add Marriage Details
2. Show Details
3. Update Details
4. Delete Details
5. Calculate Remaining Amount
6. Exit
Enter your choice: 4
Enter Bride's Name to delete details: Aурthy
Details deleted successfully!

...

```

5. Calculating Remaining Budget

The “**Calculate Remaining Amount**” feature calculates and displays the remaining budget for each wedding event by subtracting the total expense from the total budget. This information is displayed for each wedding event. Here is the relevant **code segment**:

def calculate_remaining_amount():

```

for wedding in wedding_data:
    total_budget = wedding['Total Budget']
    total_expense = wedding['Total Expense']
    remaining_amount = total_budget - total_expense
    print(f'Remaining Amount for {wedding['Groom Name']} and {wedding['Bride Name']} is:
{remaining_amount}')

```

Output:

```
Marriage Details added successfully.

Wedding Event Management System
1. Add Marriage Details
2. Show Details
3. Update Details
4. Delete Details
5. Calculate Remaining Amount
6. Exit
Enter your choice: 5
Remaining Amount for Soudeep and Arthy is: 100000.0
Remaining Amount for Biplob and Aurthy is: 200000.0
```

6. Exiting the Application

The "Exit" option allows users to exit the application gracefully. When user enter 6 then the executing program will exit.

```
while True:
    print("\nWedding Event Management System")
    print("1. Add Marriage Details")
    print("2. Show Details")
    print("3. Update Details")
    print("4. Delete Details")
    print("5. Calculate Remaining Amount")
    print("6. Exit")

    choice = input("Enter your choice: ")

    if choice == '1':
        add_marriage_details()
    elif choice == '2':
        show_details()
    elif choice == '3':
        update_details()
    elif choice == '4':
        delete_details()
    elif choice == '5':
        calculate_remaining_amount()
    elif choice == '6':
        break
    else:
        print("Invalid choice. Please try again.")
```

This is a very simple and effective project which easily manages wedding event details, including budgeting and expense tracking. Users can add, view, update, and delete wedding information, as well as calculate the remaining budget, making it a useful tool for event planning.