ASSIGNMENT 1   NAME : SNEHA HIREMATH

                                                    USN : 2GI23CS144

PROBLEM STATEMENT: Google Form: Survey

STEPS:

1. Sign in to Google

- Go to [Google Forms](#).

- Sign in with your Google account.

2. Start a New Form

- Click on the Blank form icon to start from scratch, or choose a template from the template gallery.

3. Title and Description

- Enter a title for your survey at the top.

- Add a description if needed to explain the purpose of the survey.

4. Add Questions

- Click on the Untitled Question field to type your first question.

- Select the question type from the dropdown menu (e.g., multiple choice, checkbox, short answer).

- Fill in answer options if applicable.

5. Customize Questions

- Use the icons on the right to:

    o Duplicate a question.

    o Delete a question.

    o Add a new question.

- Toggle the required question option if necessary.

6. Customize Form Settings

- Click on the Settings gear icon in the top right corner.

- Adjust settings for responses, such as collecting email addresses, limiting responses, or allowing respondents to edit their answers.

7. Theme and Design

- Click on the Palette icon to customize the theme and colors of your form.

- You can also add images or videos to enhance your survey.

8. Preview Your Form

- Click the Preview icon (eye icon) in the top right to see how your form will look to respondents.
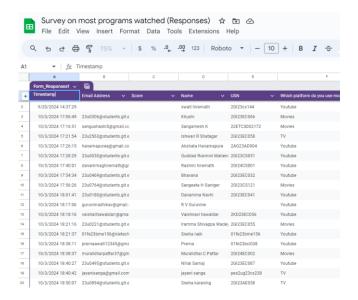
9. Send Your Survey

- Click the Send button in the top right corner.

- Choose how you want to share your survey: via email, link, or embed it on a website.
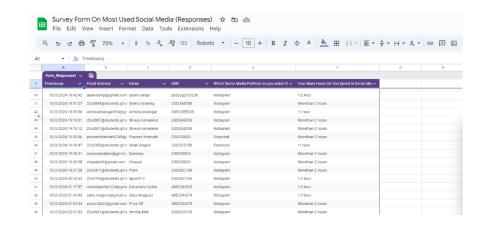
10. Collect and Analyze Responses

- After distributing your form, you can monitor responses in the Responses tab of your form.

- You can view summary charts or individual responses, and export data to Google Sheets for further analysis.

A. Create Survey Form on: Most programs watched (TV, Movies, Youtube videos, OTT, I don't watch anything)
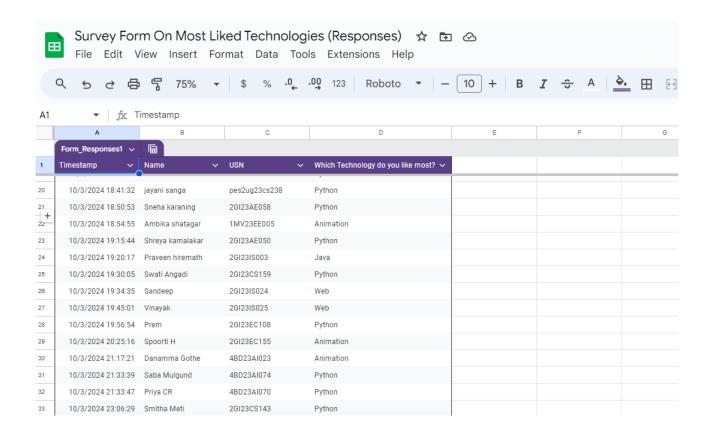
OUTPUT:

B. Create Survey Form on: Most Social Media used (Facebook, Instagram, Snapchat , Twitter, Printerest)

OUTPUT:

C. Create Survey Form on: Most liked technologies (Python, Java, Web, Animation, I don't like new technologies)

OUTPUT:

Survey Form On Most Liked Technologies (Responses)

	Timestamp	Name	USN	Which Technology do you like most?
20	10/3/2024 18:41:32	jayani sanga	pes2ug23cs238	Python
21	10/3/2024 18:50:53	Sneha karaning	2GI23AE058	Python
22	10/3/2024 18:54:55	Ambika shatagar	1MV23EE005	Animation
23	10/3/2024 19:15:44	Shreya kamalakar	2GI23AE050	Python
24	10/3/2024 19:20:17	Praveen hiremath	2GI23IS003	Java
25	10/3/2024 19:30:05	Swati Angadi	2GI23CS159	Python
26	10/3/2024 19:34:35	Sandeep	2GI23IS024	Web
27	10/3/2024 19:45:01	Vinayak	2GI23IS025	Web
28	10/3/2024 19:56:54	Prem	2GI23EC108	Python
29	10/3/2024 20:25:16	Spoorti H	2GI23EC155	Animation
30	10/3/2024 21:17:21	Danamma Gothe	4BD23AI023	Animation
31	10/3/2024 21:33:39	Saba Mulgund	4BD23AI074	Python
32	10/3/2024 21:33:47	Priya CR	4BD23AI070	Python
33	10/3/2024 23:06:29	Smitha Meti	2GI23CS143	Python

ASSIGNMENT 2

NAME : SNEHA HIREMATH

USN : 2GI23CS144

PROBLEM STATEMENT: QUIZ : One great way to learn material is to create a test or quiz over the content. Have students use Google Forms to create their own multiple-choice, True/False, fill-in-the-blank, or quizzes on the content they are learning.

STEPS:

Step 1: Access Google Forms

1. Go to Google Forms: Visit forms.google.com.

2. Sign In: If you aren't already signed in, log in with your Google account.

Step 2: Start a New Quiz

1. Create a New Form: Click on the blank form icon or select a template.

2. Turn on Quiz Mode: Click on the settings gear icon in the top right corner.

    o   Navigate to the Quizzes tab.

    o   Toggle on Make this a quiz.

Step 3: Set Up Quiz Settings

1. Choose Quiz Options: Decide if you want to release grades immediately or later, and if respondents can see missed questions, correct answers, and point values.

2. Save your settings.

Step 4: Add Questions

1. Add Questions: Click on the "+" icon to add a new question.

2. Select Question Type: Choose from multiple-choice, checkboxes, dropdowns, short answer, etc.

3. Enter Your Question: Type in your question and provide answer options (if applicable).

Step 5: Mark Correct Answers

1. Answer Key: After entering your question, click on Answer Key.

2. Select Correct Answer: Choose the correct answer(s) and assign point values.

3. Add Feedback (optional): You can provide feedback for correct and incorrect answers.

Step 6: Customize Your Quiz

1. Title and Description: Add a title and description for your quiz at the top.

2. Theme and Design: Click on the palette icon to customize the theme, colors, and fonts.

Step 7: Preview Your Quiz

1. Preview: Click on the eye icon to see how your quiz will look to respondents.

2. Test Your Quiz: Go through the quiz to ensure everything works as intended.

Step 8: Share Your Quiz

1. Send: Click on the "Send" button in the top right.

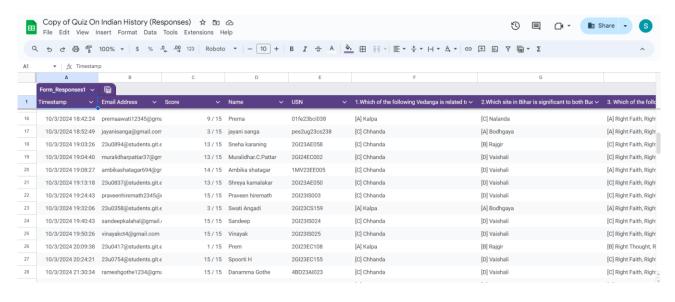2. Share Options: You can share via email, link, or embed it on a website. Adjust the sharing settings as needed.
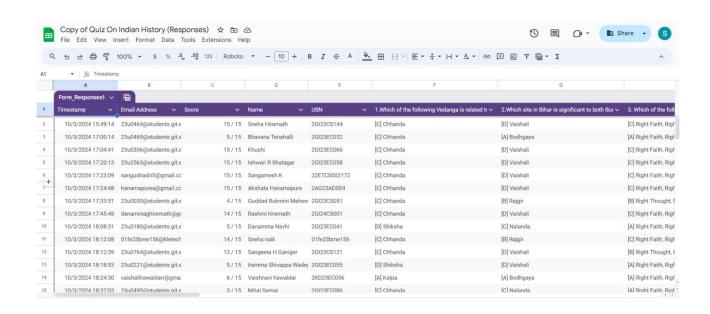
Step 9: Collect Responses

1. View Responses: As responses come in, click on the Responses tab to see the results.

2. Export Data: You can link responses to a Google Sheets document for more detailed analysis.
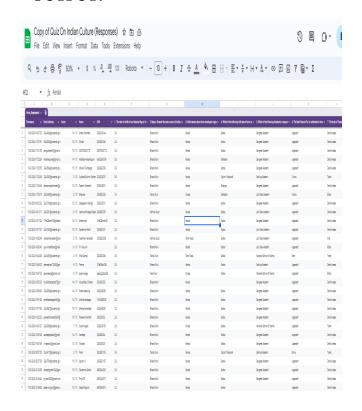
Step 10: Review and Grade Responses

1. Review Individual Responses: Check each response for grading if you're not using automatic grading.

2. Provide Feedback: Optionally, send individualized feedback or grades to respondents.

A. Create a quiz on : Indian history

OUTPUT:

B. Create a quiz on : Indian culture

 OUTPUT:

ASSIGNMENT   4   NAME : SNEHA HIREMATH

                                                               USN : 2GI23CS144

PROBLEM STATEMENT: Programs to Implement Online Voting System

ALGORITHM:

STEP 1: Initialization

- Set up data structures for users and candidates.

STEP 3: User Registration

- Prompt for username and password.

- Store the user details.

STEP 4:   User Login

- Prompt for username and password.

- Validate credentials.

STEP 5:   Add Candidate (Admin Functionality)

- Input candidate name and initialize vote count.

STEP 6: Voting Process

- Display candidates.

- Record the vote for the selected candidate.

STEP 7:   Display Results

- Show the total votes for each candidate.

STEP 8:   Exit the Program

IMPLEMENTATION LOGIC:

1. Data Structures:

- Use arrays to store user and candidate information.

- Define a structure for users and candidates.

2. Functionality:

- Implement user registration to create user accounts.

- Implement user login to authenticate users.

- Admin functions for adding candidates.

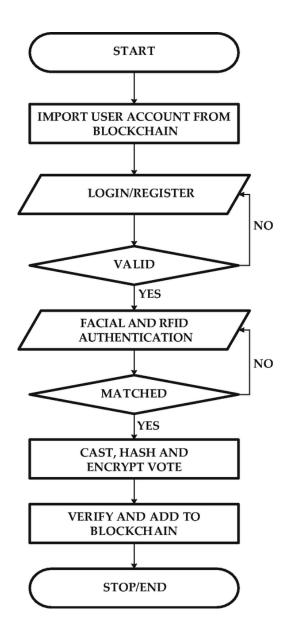- Allow users to vote, ensuring each user can only vote once.

- Display voting results after the voting process.

3. User Interaction:

- Utilize console input/output for user interaction.

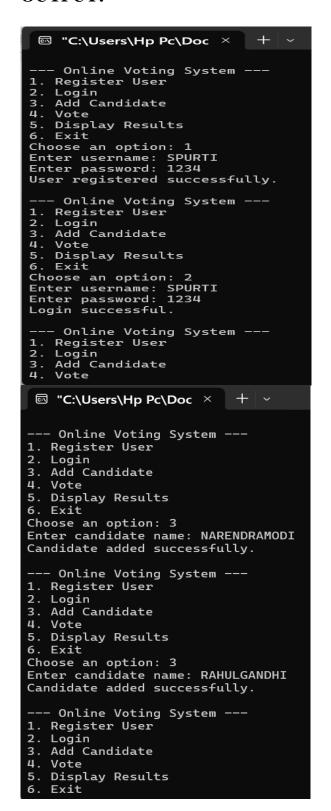Provide feedback on actions (successful registration, voting, etc.).

FLOWCHART:

```
                    ┌──────────────────┐
                    │      START       │
                    └──────────────────┘
                             │
                             ▼
              ┌───────────────────────────────┐
              │  IMPORT USER ACCOUNT FROM      │
              │        BLOCKCHAIN              │
              └───────────────────────────────┘
                             │
                             ▼
              ┌───────────────────────────────┐
              │       LOGIN/REGISTER          │◄───┐
              └───────────────────────────────┘    │
                             │                      │ NO
                             ▼                      │
                       ◇ VALID ◇──────────────────┘
                             │
                            YES
                             │
                             ▼
              ┌───────────────────────────────┐
              │     FACIAL AND RFID           │◄───┐
              │     AUTHENTICATION            │    │
              └───────────────────────────────┘    │ NO
                             │                      │
                             ▼                      │
                       ◇ MATCHED ◇────────────────┘
                             │
                            YES
                             ▼
              ┌───────────────────────────────┐
              │    CAST, HASH AND             │
              │    ENCRYPT VOTE               │
              └───────────────────────────────┘
                             │
                             ▼
              ┌───────────────────────────────┐
              │   VERIFY AND ADD TO           │
              │      BLOCKCHAIN               │
              └───────────────────────────────┘
                             │
                             ▼
                    ┌──────────────────┐
                    │    STOP/END      │
                    └──────────────────┘
```

PROGRAM:

```c
#include <stdio.h>
#include <string.h>
#define MAX_C 11

typedef struct Candidate {
        char name[50];
        int votes;
        char symbol;
} Candidate;

Candidate allCandidate
s[MAX_C];
int candidateCount = 0;
char symbols[10]
        = { '!', '@', '#', '$', '%', '^', '&', '*', '~', '+' };
int symbolTaken[11];
void fillCandidate(int);
void displayAllCandidates();
void getVotes(int);
void getResults();

int main()
{
        for (int i = 0; i < 11; i++) {
                symbolTaken[i] = 0;
        }

        printf("Enter the number of candidates: ");
        scanf("%d", &candidateCount);
        if (candidateCount >= MAX_C) {
                printf("Number of candidates cannot be greater "
                        "than 10.\n Terminating the program\n\n");
                return 0;
        }
        for (int i = 0; i < candidateCount; i++) {
                fillCandidate(i);
        }

        int numVoters;
        printf("Enter the number of voters: ");
        scanf("%d", &numVoters);

        for (int i = 0; i < numVoters; i++) {
                getVotes(i);
        }

        getResults();

        return 0;
}
void fillCandidate(int cNum)
{
        printf("Available Symbols: \n");
```

```c
            for (int j = 0; j < 10; j++) {
                    if (symbolTaken[j] == 1)
                            continue;
                    printf("%d %c\n", j + 1, symbols[j]);
            }

            int num = 0;

            printf("\nEnter the symbol number of candidate %d: ",
                    cNum + 1);
            scanf("%d", &num);

            if (num <= 0 || num > 10 || symbolTaken[num - 1] == 1) {
                    printf("This Symbol is not available. Please "
                            "choose from the available symbols\n");
                    num = 0;
                    fillCandidate(cNum);
            }
            else {
                    symbolTaken[num - 1] = 1;
                    allCandidates[cNum].symbol = symbols[num - 1];
                    printf("Enter the name of candidate %d: ",
                            cNum + 1);
                    scanf("%s", allCandidates[cNum].name);

                    allCandidates[cNum].votes = 0;
            }
    }
    void displayAllCandidates()
    {
            if (!allCandidates || !candidateCount) {
                    perror("Invalid Candidate Array\n");
                    return;
            }

            for (int j = 0; j < candidateCount; j++) {
                    printf("%s\t\t", allCandidates[j].name);
            }
            printf("\n");
            for (int j = 0; j < candidateCount; j++) {
                    printf("%3c\t\t\t", allCandidates[j].symbol);
            }
            printf("\n");
    }
    void getVotes(int voterCount)
    {
            displayAllCandidates();
            printf("Voter %d, please enter your choice (1-%d): ",
                    voterCount + 1, candidateCount);
            int choice;
            scanf("%d", &choice);
            if (choice >= 1 && choice <= candidateCount) {
                    allCandidates[choice - 1].votes++;
            }
            else {
```

```c
                        printf("Invalid choice! Please vote again.\n");
                        getVotes(voterCount);
                }
        }
}
void getResults()
{
        int maxVotes = 0;
        int winnerIndex = -1;
        int winnerFrequency = 0;
        for (int i = 0; i < candidateCount; i++) {
                if (allCandidates[i].votes > maxVotes) {
                        maxVotes = allCandidates[i].votes;
                        winnerIndex = i;
                }
        }

        for (int i = 0; i < candidateCount; i++) {
                if (allCandidates[i].votes == maxVotes) {
                        winnerFrequency++;
                }
        }

        printf("\n-----RESULT-----\n");
        if (winnerFrequency > 1) {
                printf("No candidate has majority votes\n");
        }
        else if (winnerIndex != -1) {
                printf("The winner is: %s\nCandidate Symbol: "
                        "%c\nwith %d votes!\n",
                        allCandidates[winnerIndex].name,
                        allCandidates[winnerIndex].symbol, maxVotes);
        }
        else {
                printf("No winner\n");
        }
}
```

OUTPUT:

```
--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
Choose an option: 1
Enter username: SPURTI
Enter password: 1234
User registered successfully.

--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
Choose an option: 2
Enter username: SPURTI
Enter password: 1234
Login successful.

--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
```

```
--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
Choose an option: 4
Enter username: SPURTI
Enter password: 1234
Candidates:
1: NARENDRAMODI
2: RAHULGANDHI
Enter the candidate number to vote for: 1
Vote recorded successfully.

--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
Choose an option: 5
Voting Results:
NARENDRAMODI: 1 votes
RAHULGANDHI: 0 votes
```

```
--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
Choose an option: 3
Enter candidate name: NARENDRAMODI
Candidate added successfully.

--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
Choose an option: 3
Enter candidate name: RAHULGANDHI
Candidate added successfully.

--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
```

```
--- Online Voting System ---
1. Register User
2. Login
3. Add Candidate
4. Vote
5. Display Results
6. Exit
Choose an option: 6

Process returned 0 (0x0)   execution time : 141.396 s
Press any key to continue.
```

ASSIGNMENT 5   NAME : SNEHA HIREMATH

USN : 2GI23CS144

PROBLEM DEFINATION : Programs to Implement simple calculator

ALGORITHM:

Step 1: Start.

Step 2:   Display menu options:

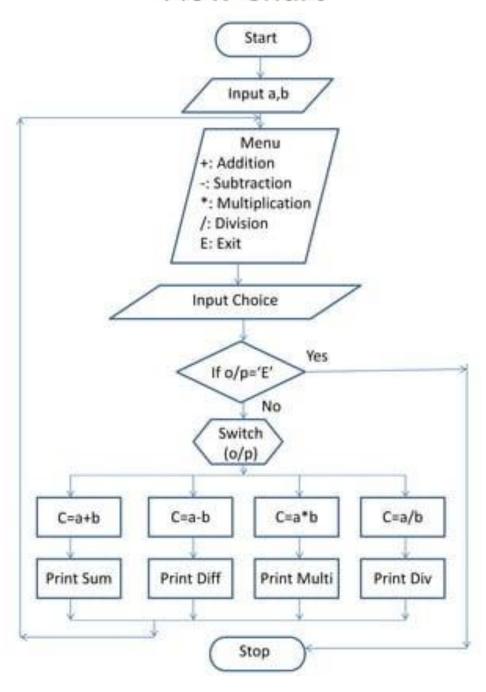- Addition
- Subtraction
- Multiplication
- Division
- Exit

Step 3:   Read the user's choice.

Step 4:   If the choice is 5, exit the program.

Step 5:   Read two numbers from the user.

Step 6:   Based on the user's choice:

- If choice is 1, add the two numbers.
- If choice is 2, subtract the second number from the first.
- If choice is 3, multiply the two numbers.
- If choice is 4, divide the first number by the second (check for division by zero).

Step 7:   Display the result.

Step 8:   Repeat from step 2 until the user chooses to exit.

Step 9: End.

IMPLEMENTATION LOGIC:

1. Menu Display: The program displays a menu with options for addition, subtraction, multiplication, division, and exit.

2. User Input: It takes the user's choice and the two numbers to perform calculations.

3. Switch Case: Based on the user's choice, the program executes the corresponding arithmetic operation using a switch-case statement.

4. Error Handling: It checks for division by zero and handles invalid choices.

5. Looping: The program continues to run until the user chooses to exit.

FLOWCHART:

Flow Chart

Start

Input a,b

Menu
+: Addition
-: Subtraction
*: Multiplication
/: Division
E: Exit

Input Choice

If o/p='E'   Yes

No

Switch
(o/p)

| C=a+b | C=a-b | C=a*b | C=a/b |

| Print Sum | Print Diff | Print Multi | Print Div |

Stop

PROGRAM :

```c
#include <stdio.h>
int main() {
    int choice;
    float num1, num2, result;
    do{
        printf("Simple Calculator\n");
        printf("1. Addition\n");
        printf("2. Subtraction\n");
        printf("3. Multiplication\n");
        printf("4. Division\n");
        printf("5. Exit\n");
        printf("Enter your choice (1-5): ");
        scanf("%d", &choice);
        if (choice == 5) {
            printf("Exiting the program.\n");
            break;
        }

        printf("Enter two numbers: ");
        scanf("%f %f", &num1, &num2);
        switch (choice) {
            case 1:
                result = num1 + num2;
                printf("Result: %.2f + %.2f = %.2f\n", num1, num2, result);
                break;
            case 2:
                result = num1 - num2;
                printf("Result: %.2f - %.2f = %.2f\n", num1, num2, result);
                break;
```

```c
            case 3:
                result = num1 * num2;
                printf("Result: %.2f * %.2f = %.2f\n", num1, num2, result);
                break;
            case 4:
                if (num2 != 0) {
                    result = num1 / num2;
                    printf("Result: %.2f / %.2f = %.2f\n", num1, num2, result);
                } else {
                    printf("Error: Division by zero is not allowed.\n");
                }
                break;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    } while (choice != 5);
    return 0;
}
```

OUTPUT:

```
 "C:\Users\Hp Pc\Doc  ×    +   ∨
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice (1-5): 1
Enter two numbers: 78 89
Result: 78.00 + 89.00 = 167.00
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice (1-5): 2
Enter two numbers: 45 56
Result: 45.00 - 56.00 = -11.00
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice (1-5): 3
Enter two numbers: 67 78
Result: 67.00 * 78.00 = 5226.00
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice (1-5): 4
Enter two numbers: 78 96
Result: 78.00 / 96.00 = 0.81
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division

Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice (1-5): 5
Exiting the program.

Process returned 0 (0x0)   execution time : 134.236 s
Press any key to continue.
```

ASSIGNMENT   7   NAME : SNEHA HIREMATH

USN : 2GI23CS144

PROBLEM STATEMENT:

All of you need to accomplish the following tasks from an individual perspective such as consider yourself to be owner of an industry, or CEO of a firm or a manager in respective field.

Create envelopes using mail merge to send out greetings to your business partners, co-workers, or employees

While creating the above mentioned see to it to include the following in your document.

·   Create your own style in MS Word.

·   Create a content Page for your Document.

·   add headers and footers to the document with proper heading styles and justification.

STEPS:

Step 1: Prepare Your Data in Excel

1.   Open Excel: Create a new spreadsheet.

2.   Enter Your Data: Include headers such as:

  o   First Name

  o   Last Name

  o   Company

  o   Address

  o   City

  o   State

  o   ZIP Code

3.   Fill in the Rows: Populate the spreadsheet with your recipients' details.

4.   Save the File: Save it as "Contacts.xlsx."

Step 2: Set Up the Word Document

Create a Content Page

1.   Open Microsoft Word: Start a new document.

2.   Insert a New Page for Content:

  o   Press Ctrl + Enter to insert a page break.

3. Create a Title:

   o Type "Mail Merge Envelopes" and format it (e.g., bold, 24 pt font).

   o Center align the title.

4. Add a Table of Contents:

   o Below the title, list the sections (e.g., "1. Prepare Data," "2. Create Envelopes," "3. Add Headers and Footers").

   o Format the list with appropriate styles (e.g., Heading 1 for main sections).

Step 3: Create the Envelope

1. Go to the Mailings Tab: Click on "Mailings" in the ribbon.

2. Select Envelopes: Click on "Envelopes" in the "Create" group.

3. Enter Return Address: Fill in your return address.

4. Click "Add to Document": This creates a blank envelope layout.

Step 4: Start Mail Merge

1. Select Recipients:

   o In the "Mailings" tab, click "Select Recipients" > "Use an Existing List."

   o Choose your "Contacts.xlsx" file.

2. Insert Merge Fields:

   o Click inside the delivery address area of the envelope.

   o Use "Insert Merge Field" to add fields like First Name, Last Name, Company, Address, City, State, and ZIP Code.

   o Format as desired (e.g., use a larger font for names).

Step 5: Add Headers and Footers

1. Insert Header:

   o Go to the "Insert" tab, click "Header," and choose a style.

   o Type the document title or your company name.

   o Format it (e.g., bold, centered).

2. Insert Footer:

   o Go to the "Insert" tab, click "Footer," and choose a style.

   o Add page numbers by selecting "Page Number" from the footer options.

   o Include the date if desired.

Step 6: Preview and Complete the Mail Merge

1. Preview Results: Click "Preview Results" in the "Mailings" tab to see how the envelopes look.

2. Finish & Merge:Choose "Finish & Merge" > "Print Documents" or "Edit Individual Documents" to create a new document.

Step 7: Print the Envelopes

1. Print Directly: If you selected to print, set your printer options and print.

2. Edit Individual Documents: If you created a new document, make final adjustments and print.

Step 8: Organize and Send

1. Sort Envelopes: Organize them as needed.

2. Deliver or Mail: Send them out to recipients.

ASSIGNMENT 8

NAME : SNEHA HIREMATH

USN : 2GI23CS144

PROBLEM STATEMENT: Demonstrate how to set the password to provide the security to your documents.

STEPS:

1. Open the Document: Open Microsoft Word and the document you wish to protect.

2. Go to File Menu: Click on the File tab in the top-left corner of the window.

3. Select Save As: Choose Save As from the menu. You can choose to save it in the cloud (OneDrive) or locally on your computer.

4. Choose the Location: Select the location where you want to save the document.

5. Open the Options: In the Save As dialog box, look for a button or link labeled Tools (this is usually located near the Save button at the bottom).

6. Select General Options: Click on Tools, then select General Options from the drop- down menu.

7. Set Passwords: In the General Options dialog box, you will see fields for:

- Password to open: Enter a password that will be required to open the document.

- Password to modify: Enter a password that will be needed to make changes to the document (optional).

  Fill in the desired password(s).

8. Confirm Passwords: After entering the passwords, click OK. You may be prompted to re-enter the password(s) to confirm.

9. Complete the Save As Process: Click Save in the Save As dialog box to save your document with the password protection.

10. Test the Password: Close the document and try reopening it to ensure that the password works as intended.

OUTPUT: