🟦 SLIDE 1: TITLE SLIDE

Auto README Generator using C Programming

Name: Gaurav Sable
Course: B.Tech
College: Rungta International Skill University
Guide: Naina Devi

🟦 SLIDE 2: AGENDA

- Introduction
- Technology Used
- Project Working
- Program Code
- Output
- Applications
- Advantages
- Conclusion

🟦 SLIDE 3: INTRODUCTION

- README file is important for GitHub projects
- Writing README manually takes time
- This project automatically generates README.md
- Developed using C programming language
- Beginner friendly and easy to understand

🟦 SLIDE 4: TECHNOLOGY USED

- Programming Language: C
- Code Editor: Visual Studio Code
- Version Control Platform: GitHub
- File Format: Markdown (.md)
- Header File Used: stdio.h

🟦 SLIDE 5: PROJECT WORKING

---

1. User runs the program
2. User enters project details
3. Program creates README.md file
4. Data is written into the file
5. File content is displayed on screen

🟦 SLIDE 6: PROGRAM CODE (OVERVIEW)

- Uses stdio.h header file
- Uses FILE pointer for file handling
- Important functions used:
  - fopen()
  - fprintf()
  - fgetc()
  - fclose()
- File is created in write mode
- File is read in read mode

🟦 SLIDE 7: PROGRAM CODE (KEY LOGIC)

---

c

Copy code

```c
file = fopen("README.md", "w");
fprintf(file, "# Project Title");
fclose(file);

file = fopen("README.md", "r");
while((ch = fgetc(file)) != EOF) {
   printf("%c", ch);
}
```

- First file is created and written
- Then same file is read and printed

```c
#include <stdio.h>

int main() {
    FILE *file;
    char title[50];
    char description[200];
    char feature1[50], feature2[50];
    char tech1[30], tech2[30];
    char ch;

    printf("===== AUTO README GENERATOR =====\n\n");
    printf("Enter Project Title: ");
    scanf(" %[^\n]", title);
    printf("Enter Project Description: ");
    scanf(" %[^\n]", description);
    printf("Enter Feature 1: ");
    scanf(" %[^\n]", feature1);
    printf("Enter Feature 2: ");
    scanf(" %[^\n]", feature2);
    printf("Enter Tech Stack 1: ");
    scanf(" %[^\n]", tech1);
    printf("Enter Tech Stack 2: ");
    scanf(" %[^\n]", tech2);

    file = fopen("README.md", "w");
    if (file == NULL) {
        printf("File cannot be created!\n");
        return 0;
    }
    fprintf(file, "# %s\n\n", title);
    fprintf(file, "## Description\n%s\n\n", description);
    fprintf(file, "## Features\n");
    fprintf(file, "- %s\n", feature1);
    fprintf(file, "- %s\n\n", feature2);
    fprintf(file, "## Tech Stack\n");
    fprintf(file, "- %s\n", tech1);
    fprintf(file, "- %s\n\n", tech2);
    fprintf(file, "## Output\n");
    fprintf(file, "README.md file ready to be pushed on GitHub\n");
    fclose(file);
    printf("\n README.md generated successfully!\n");

    file = fopen("README.md", "r");
    if (file == NULL) {
        printf("File open nahi ho pa rahi hai!\n");
        return 0;
    }
    printf("\n===== README.md FILE CONTENT =====\n\n");
    while ((ch = fgetc(file)) != EOF) {
        printf("%c", ch);
    }
    fclose(file);
    return 0;
}
```

🟦 SLIDE 8: OUTPUT

- README.md file is generated
- Output is displayed on screen
- File is properly formatted
- Ready to push on GitHub

```
===== AUTO README GENERATOR =====

Enter Project Title: README GENERATOR
Enter Project Description: first project
Enter Feature 1: a
Enter Feature 2: b
Enter Tech Stack 1: c programming
Enter Tech Stack 2: github

 README.md generated successfully!

===== README.md FILE CONTENT =====

# README GENERATOR

## Description
first project

## Features
- a
- b

## Tech Stack
- c programming
- github

## Output
README.md file ready to be pushed on GitHub
PS C:\Users\hp\OneDrive\Desktop\Gaurav sable (10519) PFC\Gaurav_ka_project>
```

🟦 SLIDE 9: APPLICATIONS

- Useful for students and beginners
- Helpful for GitHub projects
- Saves time in documentation
- Used in college mini projects
- Can be extended for future use

🟦 SLIDE 10: ADVANTAGES

- Simple and easy project
- Beginner friendly
- Uses basic C concepts
- Automates README creation
- Improves documentation skills

🟦 SLIDE 11: CONCLUSION

- Auto README Generator is a useful C project
- Demonstrates file handling concepts
- Helps understand real-world programming
- Good learning experience for beginners
- Can be improved in future