

CSE 1320 Project Documentation

## [SAFE EXIT](https://www.studocu.com/in/course/ballari-institute-of-technology-and-management/network-theory/5842786?utm_campaign=shared-document&utm_source=studocu-document&utm_medium=social_sharing&utm_content=c-programming-full-project-documentation)

## Students names, surnames, IDs:

## 1.Ayush Adhikari, 1002095135

## 2.Diyanshu Naag Goli 1002223126

## 3.Anil Ghimire 1002231127

## 4.Vedant Bhalke 1002204217

Mentor: Marika Apostolova

**PROJECT LOGO**

**Intermediate** **programming** **CSE 1320**

**Student** **declaration:**

*We declare* *that:*

* *We* *understand* *what* *is* *meant* *by* *plagiarism*
* *The* *implication* *of* *plagiarism* *has* *been* *explained* *to* *me* *by* *our* *professor*
* *This* *assignment* *is* *all* *team* *own* *work* *and* *we* *have* *acknowledged* *any* *use* *of* *the* *published* *and* *unpublished* *works* *of* *other* *people.*

**1 Student** Ayush Adhikari, 1002095135 **signature:** Ayush

**2 Student** Anil Ghimire 1002231127 **signature:** Anil

**3 Student** Diyanshu Naag Goli 1002223126 **signature:** Divyanshu

**4 Student** Vedant Bhalke 1002204217 and **signature:** Vedant

**Date:4/20/2025**

|  |  |  |
| --- | --- | --- |
|  | **Total** **number** **of** **pages** **including** **this** **cover** **page** | 7 |
| **Class** **Code** **/** **Group** | CSE 1320 | |
| **Lecturer’s** **Name** | MARIKA APOSTOLOVA | |

**Table of Contents**

[**1. Project Introduction 1**](#_Toc174624498)

[**2. Project Description 2**](#_Toc174624499)

[**3. 1**](#_Toc174624500)

[**4. 1**](#_Toc174624501)

1. **Project Introduction**

The *SafeExit* project is designed to simulate a building evacuation assistant system. It allows users to input a room number and get the safest and shortest exit route. The program reads room-to-exit path data from a file and outputs appropriate directions, helping users navigate emergency exit plans effectively.

1. **Project Description**   
   **Goals and Objectives:**

To create an emergency routing system using C that aids in fire or disaster evacuation.

To practice file handling and user interaction through the command line.

To implement route lookup logic using simple search and matching techniques.

**Need for the Project:** In real-world emergencies, knowing the nearest safe exit route can save lives. This project mimics such a system using a text-based simulation which could be foundational for real building safety applications.

**Phases:**

**Data Input:** Load room and route data from routes.txt.

**User Interaction:** Take user input for room queries.

**Search Logic:** Find corresponding exits.

**Output:** Display path or help instructions.

**Exit:** Allow graceful termination of the program.

1. **Project Architecture**

**A diagram of a computer program

AI-generated content may be incorrect.**

1. **Programming Concepts Used**

 **File Handling**: The program opens routes.txt and reads mappings using fopen, fgets, and string parsing.

 **Strings and Parsing**: strcmp, strtok, and manual string comparisons help match room names.

 **Loops and Conditionals**: Used to iterate through file content and determine logic branches.

 **Functions**: strcasecmp (case-insensitive match), string utilities, and helper logic.

 **CLI Interaction**: Uses scanf, printf, and command line arguments to manage interaction.

1. **Code Description**

#### ****Feature 1: File Reading****

FILE \*fp = fopen(argv[1], "r");

while (fgets(line, sizeof(line), fp)) {

// Read each line from routes.txt

}

**Description**: Opens the file and reads each line into a buffer.

#### ****Feature 2: Case-Insensitive Search****

if (strcasecmp(input, "HELP") == 0) {

// Display all rooms and exits

}

**Description**: Ensures user commands are interpreted correctly regardless of case.

#### ****Feature 3: Route Lookup****

if (strstr(line, input)) {

printf("Escape Route: %s", line);

found = 1;

}

**Description**: Searches for the user-provided room number and outputs the escape route.

1. **System Testing**

A screenshot of a computer

AI-generated content may be incorrect.

1. **Group members**

Ayush Adhikari – Worked on the project documentation, including the summary, technologies, and user guide. Also helped test the program for edge cases and provided feedback on usability.

Anil Ghimire - Led the programming and implementation of the main C code, handled file reading and logic for room-to-exit mapping, and debugged runtime issues.

Diyanshu Naag Goli - Designed the structure of the routes.txt file and contributed to implementing the HELP and QUIT features.

Vedant Bhalke - Assisted with code comments, compiling and running the program on different systems, and final report formatting and submission.

1. **Conclusion and future works**

The *SafeExit* project effectively demonstrates a basic implementation of an emergency route system using C programming fundamentals. It incorporates file handling, user interaction, and search logic to produce an interactive CLI tool.

**Future Work:**

* Add GUI elements to improve accessibility.
* Include route validation and error correction.
* Expand route file format with map-like logic or graphs.
* Integrate with sensors or real-time alerts.