# A black and red text Description automatically generated

**PROJECT AND TEAM INFORMATION**

## Project Title

|  |
| --- |
| Lost and Found portal for college using C |

## Student/Team Information

|  |  |
| --- | --- |
| Team Name: | Byte Storm |
| Team member 1 (Team Lead)  ( Negi : Antriksh : 241601259: [antrikshnegi2007@gmail.com](mailto:antrikshnegi2007@gmail.com) ) |  |
| Team member 2  ( Tiwari : Ayush : 241601311 : [ayush.tiwari0618@gmail.com](mailto:ayush.tiwari0618@gmail.com) ) | A person wearing glasses and a white shirt  AI-generated content may be incorrect. |
| Team member 3:  ( Thapa : Kriti : 241602072 : [kritithapa609@gmail.com](mailto:kritithapa609@gmail.com) ) |  |
| Team member 4 :  ( Chauhan : Rohit : 241601103 : [Rc9634708@gmail.com](mailto:Rc9634708@gmail.com) ) |  |

**PROJECT PROGRESS DESCRIPTION (35 pts)**

## Project Abstract (2 pts)

|  |
| --- |
| The **Lost and Found Portal for College** is a C-based system designed to assist students and faculty in reporting line and retrieving lost or found items on campus. It simplifies the process by providing a command- menu interface, where users can register lost/found items, search for them, request contact information, and notify admins. The system uses persistent file storage for saving data and includes categorized entries, date records, admin login, and contact request logs. In the final phase, an **Auto Hot key-based executable launcher (.exe)** was created to simplify launching the main program without using the command prompt, improving usability for all users. |

## Updated Project Approach and Architecture (2 pts)

|  |
| --- |
| The project follows a modular C programming structure, with separate functions for item registration, search, updates, file management, and admin operations. It uses libraries like stdio.h, stdlib.h, string.h, time.h, and ANSI color codes for a better CLI experience. Data persistence is handled through text files (items.txt, notifications.txt, contact\_requests.txt), and the program was compiled using **Code::Blocks MinGW compiler** into an .exe file for deployment. **Additionally, a compiled Auto Hotkey (.ahk) script created a GUI launcher executable to run the main C program’s .exe via a simple button interface**, making it accessible for non-technical users. |

## 

## Tasks Completed (7 pts)

|  |  |
| --- | --- |
| Task Completed | Team Member |
| * Designed and finalized system structure      * Implemented item registration, search, and display modules * Developed file handling and admin operations (delete/view logs) * Added notifications, contact request, and admin login * Compiled final .exe file from C program * Created Auto Hot key launcher executable (.exe) | Antriksh Negi  Rohit Chauhan  Ayush Tiwari  Kriti Thapa  All  Antriksh Negi |

## Challenges/Roadblocks (7 pts)

|  |
| --- |
| Challenges included managing file corruption risks during unexpected exits, handling duplicate entries, validating varied user inputs, and securing admin access. Integrating a clean CLI user interface with color-coded prompts was time-intensive. During Phase 3, ensuring compatibility of the compiled .exe across different systems and integrating it with the AutoHotkey launcher required careful testing and iteration. These were resolved through input validation improvements, unique entry checks, iterative test cycles, and GUI enhancements in the launcher .exe to ensure seamless user experience. |

## Tasks Pending (7 pts)

|  |  |
| --- | --- |
| Task Pending | Team Member (to complete the task) |
| Final documentation and user manual  Optional: Backup file creation for recovery (future enhancement) | All  All |

## 

## Project Outcome/Deliverables (2 pts)

|  |
| --- |
| The final project deliverables include:   * **Lost and Found portal executable (.exe)** compiled from C program. * Source code with modular implementation. * Persistent file storage (items.txt, notifications.txt, contact\_requests.txt). * Admin functionalities (delete items, view notifications, contact requests). * **AutoHotkey launcher executable (.exe)** for easy program startup. * Testing reports. * Final documentation (in progress). |

# Progress Overview (2 pts)

|  |
| --- |
| **100% of core development is complete.** All planned modules and features have been implemented, tested, and integrated. Both the **C program executable** and the **AutoHotkey launcher executable** have been generated and validated. Minor documentation tasks and final polish remain. No significant delays encountered in the final phase. |

# Codebase Information (2 pts)

|  |
| --- |
| Git Hub Accessible Link: https://github.com/AntrikshNegi001/PBL001.git  Branch : Main  1 parent : bf523a0  2 commit :1c5881b |

## Testing and Validation Status (2 pts)

|  |  |  |
| --- | --- | --- |
| Test Type | Status (Pass/Fail) | Notes |
| Unit Testing  Integration Test  Error Handling | Pass  Pass  Pass | All individual modules tested and working as expected.  Modules integrated successfully and functioning well.  Input validation, admin login, and file operations tested |

# Deliverables Progress (2 pts)

|  |  |
| --- | --- |
| Deliverable:  * Search and Match System * Item Registration Module * File Storage and Retrieval * Contact Linking Feature * Admin Functionalities (Delete/View Logs) * Final Interface (CLI-based Menu) * Final Executable File (.exe) (C Program) * AutoHotkey Launcher Executable (.exe) * Testing and Documentation | Status: Completed  Completed  Completed  Completed  Completed  Completed  Completed  Completed  Completed |