**1. TOPIC (Title Page)**

Project name: Historical Events Management System

Team name: Code Titans

|  |  |
| --- | --- |
| Egor Grebennikov | Scrum Trainer |
| Alexandar Stanev | Developer |
| Kristian Rusev | Developer |
| Gabriel Yanchev | Developer |
| Tsvetomir Stoilov | QA |

**3. ABSTRACT**

**3.1 Objectives**

The project aims to provide a C++ application for managing historical events, with a focus on Bulgarian history. It allows adding, searching, editing, and filtering events using a **doubly linked list**. While most similar tools exist online or in academic books, this is a lightweight, local solution implemented in C++.

**3.2 Key Implementation Stages**

* Planning using SCRUM methodology
* Developing structure and code
* Testing
* Preparing documentation and presentation

**3.3 Implementation – Technologies Used**

* Programming Language: C++
* Data Structure: Doubly Linked List
* IDE: Visual Studio
* Version Control: GitHub
* Documentation: MS Word

**3.4 Application Description (with diagram)**

A diagram of a function

AI-generated content may be incorrect.

**4. TECHNICAL CONTRIBUTIONS   
4.1 Purpose and Intended Use**The application is intended to manage historical events by allowing users to add, edit, search, and filter them in a local environment. It demonstrates core programming concepts such as dynamic memory management, data structures, and object-oriented design in C++. The system is useful for students learning C++ and for anyone needing a simple way to organize historical information.

**4.2 Team Responsibilities and Contributions**

**a) Scrum Trainer – Egor Grebennikov**

* Set up the GitHub repository and initialized the project structure
* Assigned roles and tasks based on the project timeline
* Organized team meetings and ensured deadlines were met
* Maintained team coordination and tracked progress using SCRUM methodology

**b) Developer – Alexandar Stanev**

* Implemented the console-based menu and user interaction logic
* Handled input/output operations for adding and displaying events
* Integrated the user interface with the core application logic

**c) Developer – Kristian Rusev**

* Developed the Event class structure including attributes like date, topic, leader, etc.
* Wrote the logic for adding, editing, deleting, and searching for events
* Supported implementation of filters and sorting functionality

**d) Developer – Gabriel Yanchev**

* Created the LinkedList class using a doubly linked list
* Managed memory and pointer operations for event storage
* Implemented sorting algorithms and ensured correct data traversal

**e) QA Engineer – Tsvetomir Stoilov**

* Created a test plan and test cases for all application features
* Performed manual testing including boundary and edge cases
* Wrote automated unit tests using assert statements
* Reported and documented bugs and suggested improvements

**5. CONCLUSION**

* This project demonstrates fundamental data structure use in C++, handling user input, and structuring a console application. The application is educational and ideal for learning dynamic memory management.

**6. TABLE OF CONTENTS**

1. Topic .................................................. 1

2. Authors ................................................. 1

3. Abstract ................................................ 1

4. Technical Contributions ................................. 2

5. Conclusion .............................................. 3

6. Table of Contents ....................................... 3