Proposal for:

Password Manager API using C++

Submitted By:

Safal Gautam *(PUR081BCT066)*

Shekhar Limbu *(PUR081BCT083)*

Submitted To:

Er. Tantra Nath Jha

Tribhuvan University
Institute of Engineering (IOE)

Department of Electronics and Computer Engineering
Purwanchal Campus
Dharan, Nepal

Date: 2082-05-03

1. Acknowledgment

I would like to express my heartfelt gratitude to my respected subject teacher, Mr. Tantra Nath Jha, whose invaluable guidance, constructive feedback, and constant support have been instrumental in helping me conceptualize and develop this project. His insightful suggestions and encouragement motivated me to approach the work with clarity and determination.

I would also like to extend my sincere appreciation to my friends and seniors, who generously shared their valuable guidance throughout the initial stages of this work. Your support and encouragement have been truly invaluable.

2. Abstract

This project presents a lightweight web-based API for a simple password manager implemented in C++ using the Drogon framework. The API allows users to create accounts, add, view, update, and delete passwords through RESTful HTTP endpoints. Passwords are stored in encrypted form using a straightforward XOR-based encryption, and user data is persisted in JSON files to simplify storage and maintenance.

The focus of this system is **simplicity**, **educational value**, and **ease of use** rather than enterprise-level security features. By providing a fully functional yet easy-to-understand solution, this project serves as a practical learning tool for students and developers exploring web APIs, encryption, and C++ programming.

Table of Contents

1. Acknowledgment 2
2. Abstract 3
3. Introduction 5
4. Objectives 5
5. Existing System 6
    5.1. Limitations of Current Solutions 6
6. Proposed System 6
    6.1. Key Features 6
    6.2. Technical Architecture 6
7. Description 7
    7.1. System Components 7
    7.2. Data Flow 7
8. System Block Diagram 8
9. Methodology 9
    9.1. Development Approach 9
    9.2. Testing Strategy 9
10. Project Scope 9

3. Introduction

The need for secure password management is becoming increasingly crucial as users maintain multiple accounts online. This project aims to provide a lightweight, developer-friendly API for storing and retrieving passwords securely.

Unlike complex enterprise-grade solutions, this project focuses on **simplicity** and **ease of learning**. The system is implemented in **C++ using the Drogon web framework**, making it suitable for educational purposes and small-scale deployments.

4. Objectives

- To build a simple REST API for password storage and retrieval
- To store passwords in encrypted form using basic XOR-based encryption
- To allow account creation and password management via HTTP endpoints
- To keep the system lightweight and easy to understand for learners

5. Existing System

5.1. Limitations of Current Solutions

- Most existing password managers are too complex for beginners to understand
- Closed-source tools limit transparency
- API-based password managers are rare and difficult to set up for students
- Most require browser extensions or mobile apps

6. Proposed System

6.1. Key Features

- Account creation with unique API keys
- Add, view, update and delete passwords via RESTful endpoints
- Passwords stored in encrypted form (XOR-based for simplicity)
- Simple JSON file-based storage instead of complex databases

6.2. Technical Architecture

The system employs a two-tier architecture:

1. Application Layer – Drogon-based C++ server handling HTTP requests
2. Data Layer – JSON file storage with encrypted passwords

7. Description

7.1. System Components

- API Controller: Handles HTTP routes for create, add, update, view, and delete operations
- Encryption Module: Provides simple XOR encryption and decryption
- JSON Storage: Saves user accounts and passwords
- Homepage/Landing Page: Displays usage instructions for the API at the root URL

7.2. Data Flow

1. User creates an account and receives an API key
2. API key is used to add and retrieve passwords
3. Passwords are stored in encrypted form in a JSON file
4. View requests return decrypted passwords to the user

8. System Block Diagram

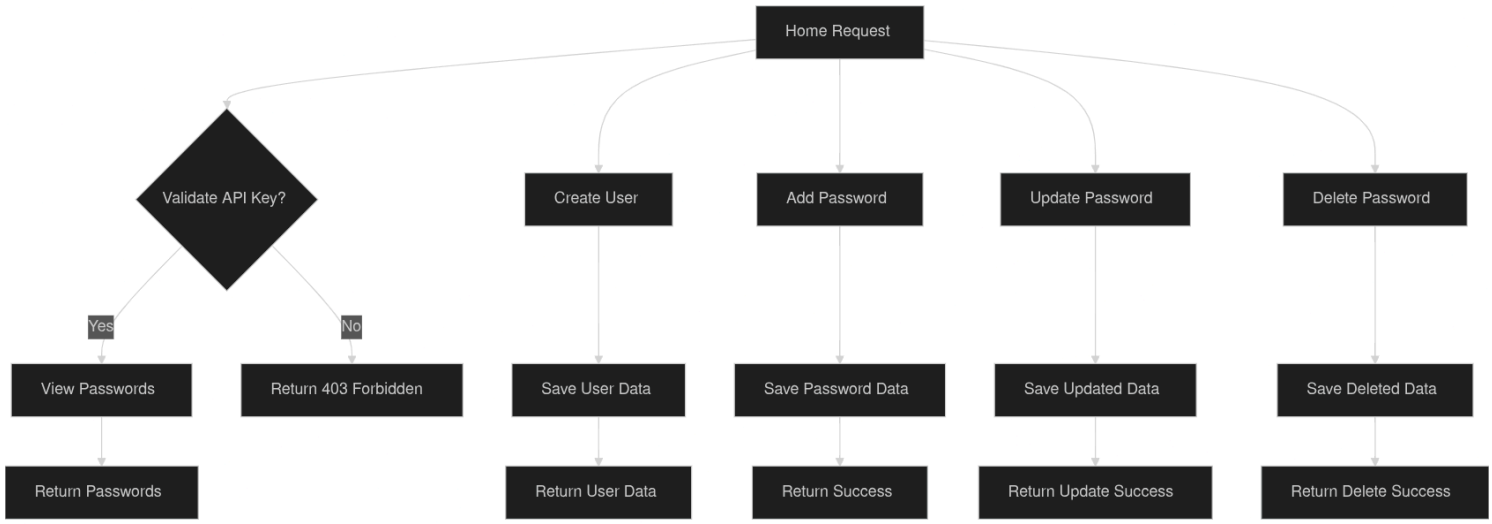

Figure 1: System block diagram

9. Methodology

9.1. Development Approach

- Plain C++ with the Drogon framework
- Step-by-step incremental development for easier understanding
- Manual testing using **cURL** commands
- File-based data persistence

9.2. Testing Strategy

- cURL-based endpoint testing
- Manual verification of JSON file changes
- Basic error handling checks

10. Project Scope

- Working Drogon-based REST API
- Instruction page at root URL (/)
- Password storage in JSON file with encryption
- Full source code provided for learning purposes