**Project Report: Object-Oriented Crypto Trading Platform**

**Group Members:**

Asghar Ali 22k-4415

Hafiz Abdullah 22k-4489

Muhammad Bilal 22k-4242

**Introduction:**

The objective of this project is to develop a crypto trading platform that uses object-oriented programming (OOP) concepts to implement its functionality. The platform will provide users with the ability to buy and sell cryptocurrencies, view their account balances, and manage their portfolios. The platform will be designed using the principles of OOP, including **encapsulation**, **inheritance**, and/or **polymorphism**, to create a flexible and scalable project. The project will be implemented using C++ programming language, and data will be stored in **.txt** files.

**Class Design:**

The crypto trading platform will be designed with the following classes:

User: This class will represent a user of the platform. It will contain attributes such as the user's name, email address, account balance and generic data types relevant for a user class (full list of attributes in the **UML class diagram** below). This class is the parent class for Account setting and crypto classes. For Methods in this class will include functions to buy and sell cryptocurrencies, view account balances, and manage portfolios.

Account setting: This class will represent a user's portfolio of cryptocurrencies. It will contain attributes such as the user's holdings of each cryptocurrency and the total value of their portfolio. Methods in this class will include functions to change attributes, add balance to the portfolio, view the portfolio history, and view the user's holdings of each cryptocurrency.

Crypto: This class will represent a cryptocurrency, with attributes such as the cryptocurrency's name, symbol, and current price. Methods in this class will include functions to update the cryptocurrency's price, view the current price, and calculate the value of a given quantity of the cryptocurrency.

**Implementation:**

The crypto trading platform will be implemented using C++ programming language. The User, Account setting, Crypto classes will be defined in C++, and files will include user information, cryptocurrency information, and order information. The C++ code will use file input and output (fstream) operations to interact with the .txt files.

The platform will use a command-line interface for user interaction. The command-line interface will allow users to create accounts, log in, view their account balances and portfolios, and place orders to buy and sell cryptocurrencies. The C++ code will handle user input and output to the console.

Conclusion:

In conclusion, this project will develop a crypto trading platform that uses OOP concepts to implement its functionality. The platform will use a command-line interface for user interaction, with the C++ code handling user input and output to the console. The project will provide a practical example of how OOP concepts can be used to design and implement a flexible and scalable system.

|  |  |
| --- | --- |
| User | |
| * firstName: string * lastName: string * username: string * email: string * phoneNumber: string * password: string * creditCardNumber: string * expiry: string * cvv: string * balance: double | |
| * logged\_in: bool * User() * is\_logged\_in(): bool * get\_first\_name(): string * set\_first\_name(firstName: string): void * set\_last\_name(lastName: string): void * set\_username(username: string): void * set\_email(email: string): void * set\_phone\_number(phoneNumber: string): void * set\_password(password: string): void * set\_credit\_card\_number(creditCardNumber: string): void * set\_expiry(expiry: string): void * set\_cvv(cvv: string): void * set\_balance(balance: double): void * get\_password(): string | * **get\_last\_name(): string** * **get\_username(): string** * **get\_email(): string** * **get\_phone\_number(): string** * **get\_credit\_card\_number(): string** * **get\_expiry(): string** * **get\_cvv(): string** * **get\_balance(): double** * **login(): string** * **signup(): void** * **load\_account\_info(username: string): void** * **view\_account\_info(): void** * **edit\_account\_info(): void** * **delete\_account(): void** |

|  |
| --- |
| Account\_setting |
| * file\_name: string |
| * change\_password(filename: string): void * change\_name(filename: string): void * change\_email(filename: string): void * change\_phone(filename: string): void * add\_balance(filename: string): void * view\_history(string): void * display\_data(string): void |

|  |
| --- |
| Crypto |
| * name: string * symbol: string * price: double |
| * Crypto(n: string, s: string, p: double) * get\_name(): string * get\_symbol(): string * get\_price(): double |