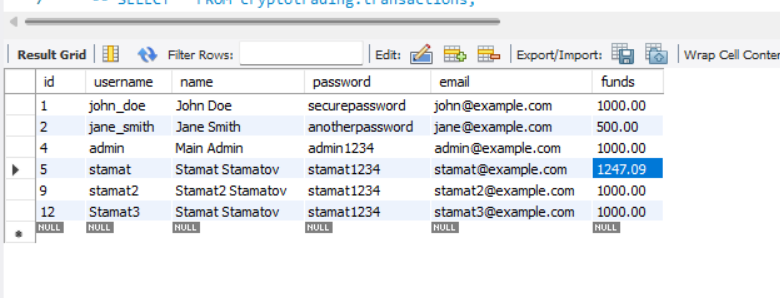
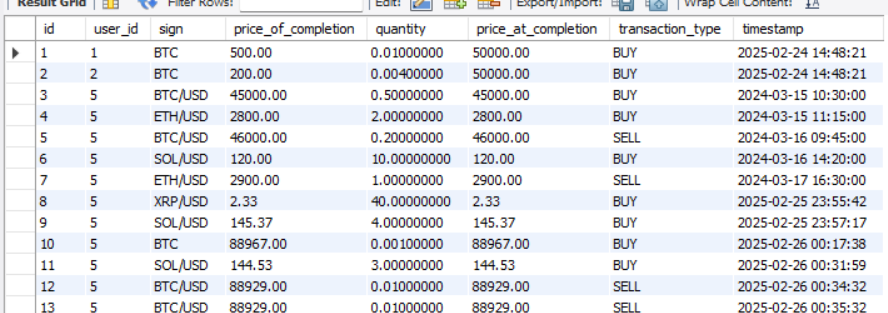
Trading Platform Documentation

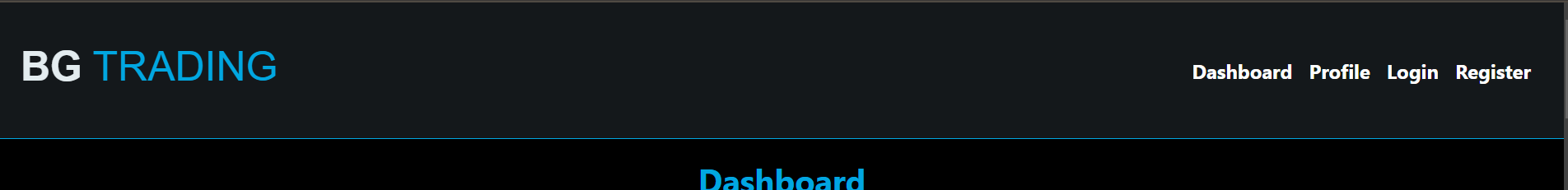
This document contains the brief documentation for the Trading Platform project.

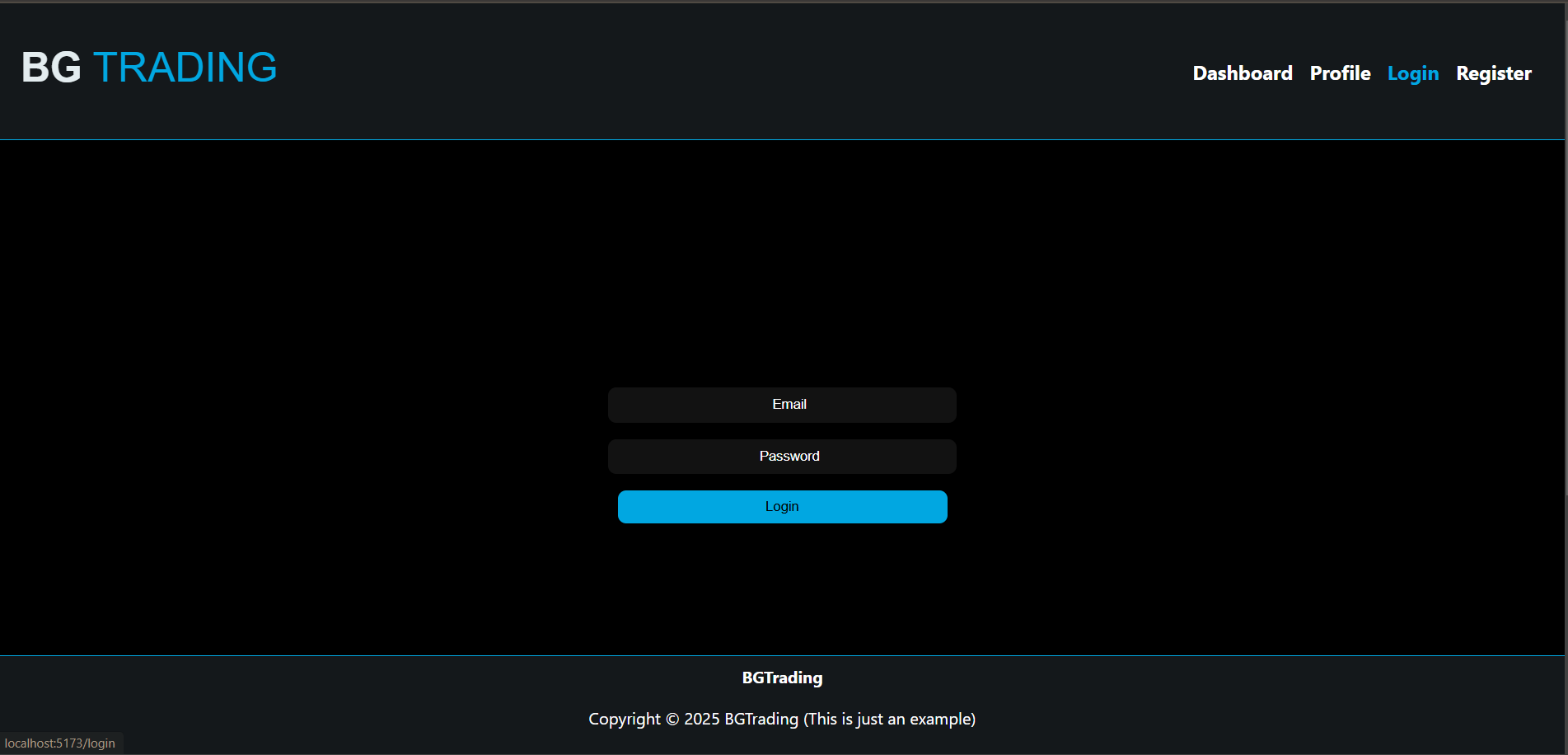
***1.Screenshots***

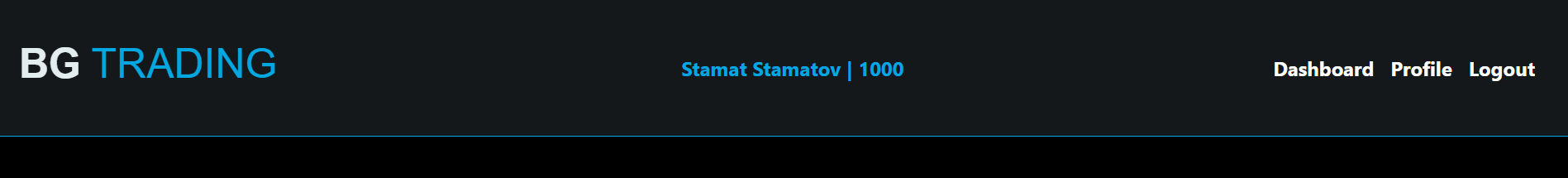


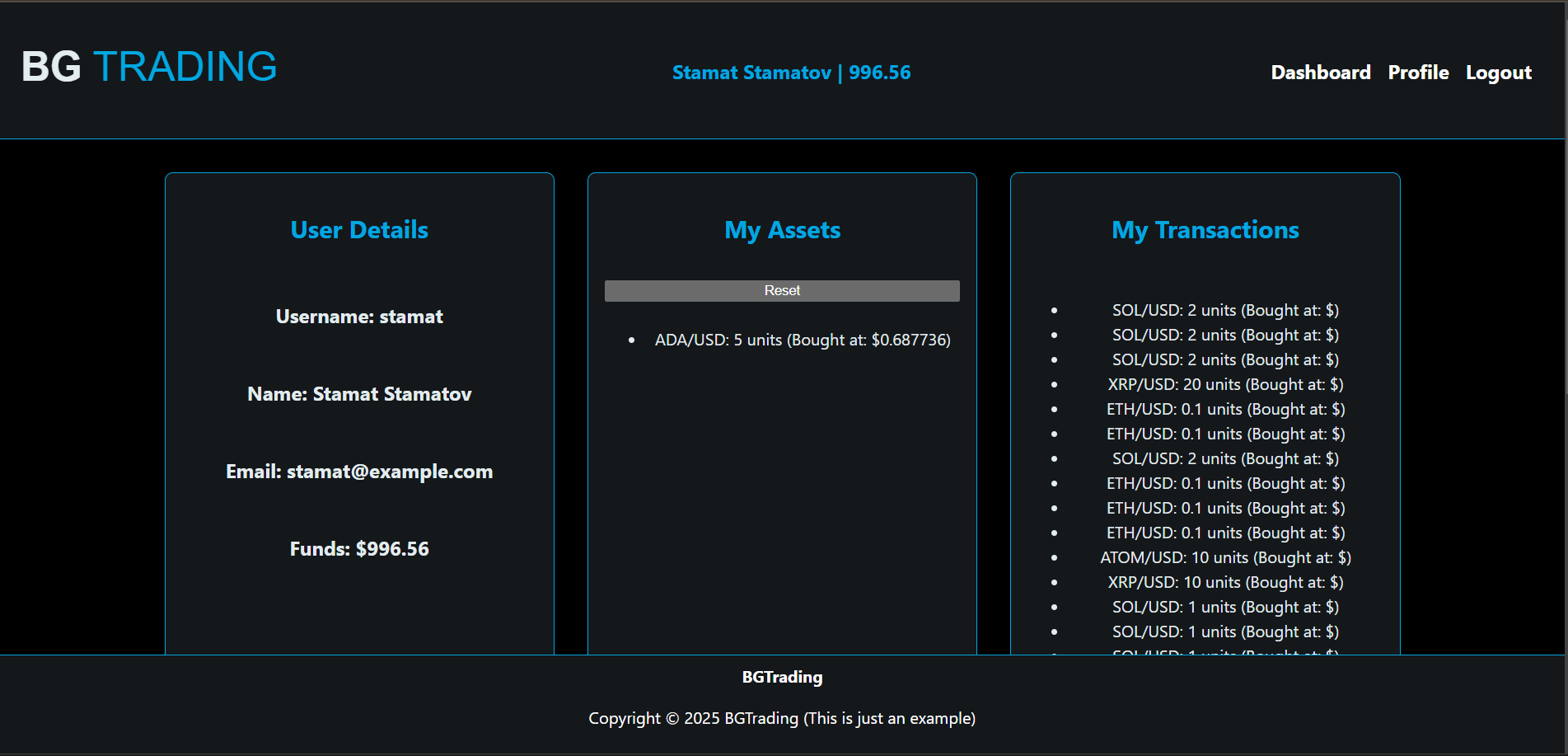


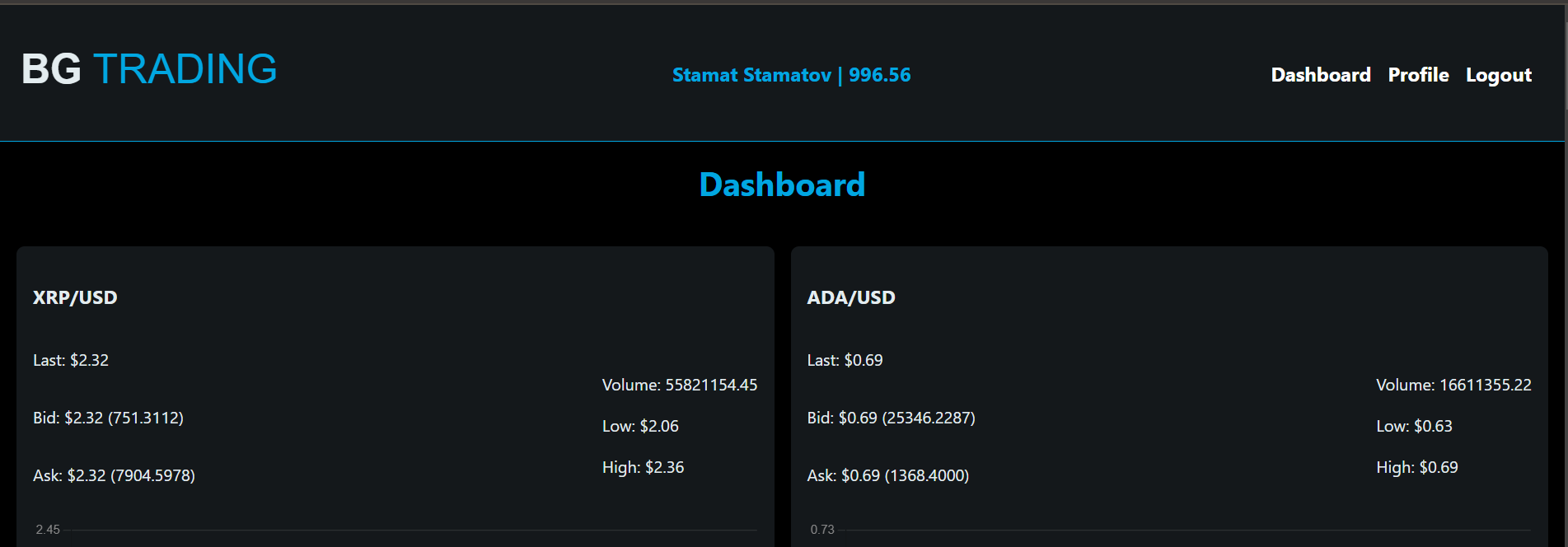
******

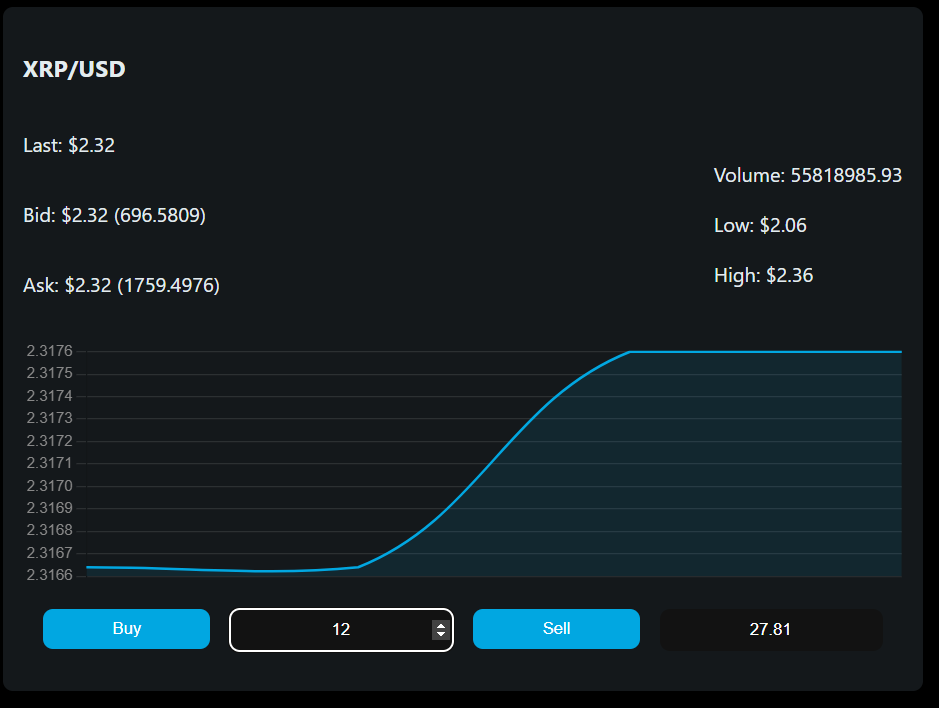
******

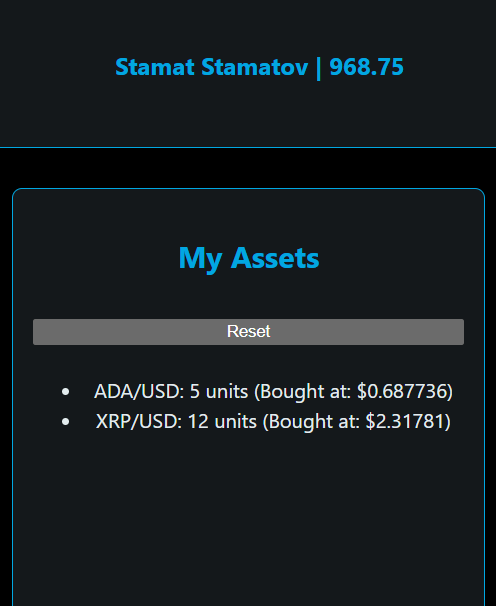
******

******

******

******

******

******

***2.Explanation of implementation***

This is a brief explanation on the whole project.

The languages, frameworks and libraries used are the following: MySQL database, Java, Spring, HTML, CSS, JavaScript, React.js, Chart.js.

I have used React.js so the DOM manipulations could be faster and I wouldn’t have to deal with the need of refreshing the whole page for each small change. This way there can be fast updating charts and crypto data.

I used Chart.js for my graphs because it is open-sourced, free and easy to use.

I created the database manually because it is small enough and I decided that an orm (or in our case jpa configuration for each class), would not be needed. In java, I created a class for each table of the datatbase. For each class I created a service, so there the functions for connecting with the database can stay. After that I created a controller to listen on the port for calls. This way the React front-end and the Spring back-end work together.

By doing so, I have a website that is changing fast and dynamically. It has fast transactions between the layers and two layers of security. (Validation in both the front and back-end.)

***3.Future plans***

For the future I would like to make the charts pre-load with data save from the api. This can be achieved simply if I use an API that sends data saved for the past. The other way around is to have a database with the previous data we need for the chart.

I would like to make the code better for reading, because I did not have enough time to do so.

Also, for a production-grade website I need to hash the user passwords.

I would also need to host the website with the back-end and database on server that can handle high user load.

A good thing would be to check the capacity that the database can hold and how many transactions pes second it can handle.

Another useful thing would be for the database to be able to handle history of the crypto prices for a period of time. This way the website would be more useful.

Lastly I would like to create a better design and better message handeling for the user.

Thanks for the time spent,

Dimo Dimitrov.