Bricked Up

Francesco Schenone   Sebestyen Deak   Leonardo Gianola   Kacper Grzyb

Ignad Bozhinov

17-12-2024

1   Introduction

The LEGO buying and selling market has become one of the most profitable businesses in recent years, with an average return of 11% [1]. These returns have sparked the interest of many people in investing in this alternative asset. However, the current process for analyzing the history, prices, trends, and news about different LEGO sets remains tedious, fragmented, and inefficient.

To solve this problem, we have developed *Bricked Up*, a software designed to centralize all the key information about a large number of LEGO sets. Our goal is to make it easier for users to make informed decisions when investing in this asset, optimizing access to relevant data and analytical tools.

Bricked Up's most relevant features include interactive graphics to visualize relevant data, a historical price comparison to evaluate the evolution of the sets, and an exploration page to discover new opportunities in a wide variety of LEGO sets.

This report details the development of the software, from its Bloomberg Terminal-inspired aesthetic to the efficient management of LEGO set data and a secure authentication system. With this project, we seek to offer an original tool that allows users to invest safely and informally in the LEGO market, democratizing access to this lucrative alternative asset.

2   Front-End

This project's first task is to build your application's front-end side. This section should clearly describe the technical implementation of the work put into building the front-end:

- Technically describe the use of HTML 5: which HTML tags do you use, where, and why

- Technically describe the use of CSS: why and how you use CSS (including interesting selectors/declarations and how it is incorporated in the application)

- Technically describe the use of JavaScript: why and how you use it in your application (including interesting behaviors and how they are incorporated into the application)

Resources. Lectures 1 to 3.

Length. 2 columns.

3   Resource Management

The second task of this project is to build a resource management. A resource is a model of an object in your system, and it could be anything: movies, music albums, pets, etc. This section should clearly describe the management of the chosen resource: the CRUD (Create-Read-Update-Delete) operations associated with that resource:

- Technically describe the resource (ie, the model)

- Technically describe how the CRUD operations are implemented

Resources. Lectures 4 to 6.

Length. 2 columns.

4 Authentication and Authorization

This project's third and final task is to incorporate authentication and authorization capabilities. This section should clearly describe:

- Authentication: The different users of the system and how it is implemented

- Authorization: Summarize the access of the different users in the system and how it is implemented

- Role table: Include a role table associating actions over the system (you can think of them as use cases) and users that can perform these actions.

Resources. Lecture 7.

Length. 2 columns.

5 Conclusions

The goal of the conclusion is similar to the introduction: it summarizes the work itself and the takeaways a reader should take when reading this work. However, it can use the information presented in the work to be more specific than the introduction.

In the context of the Web Technologies course, the conclusion should clearly describe:

- Summary: summary of the work and main takeaways. Also include a class diagram of the system (the models).

- Future Work: interesting directions on how the presented work can evolve in the future (it may be the starting point to choose individual extension topics)

Length. Half a column.

References

[1] Dmitry B. Krylov, "LEGO investing as an alternative asset class: Annualized returns of 11%," *ScienceDirect*, 2021. Available at: ScienceDirect Article.