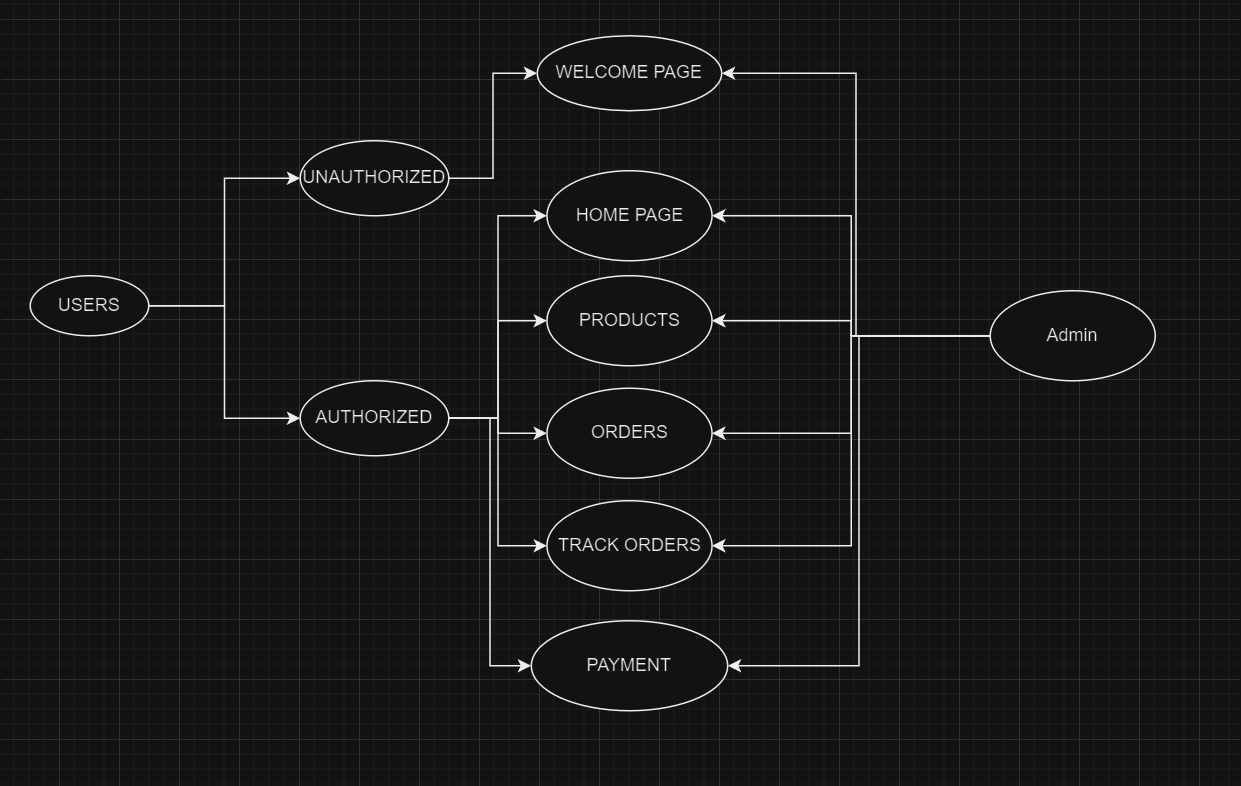
**TOY STORE APPLICTION**

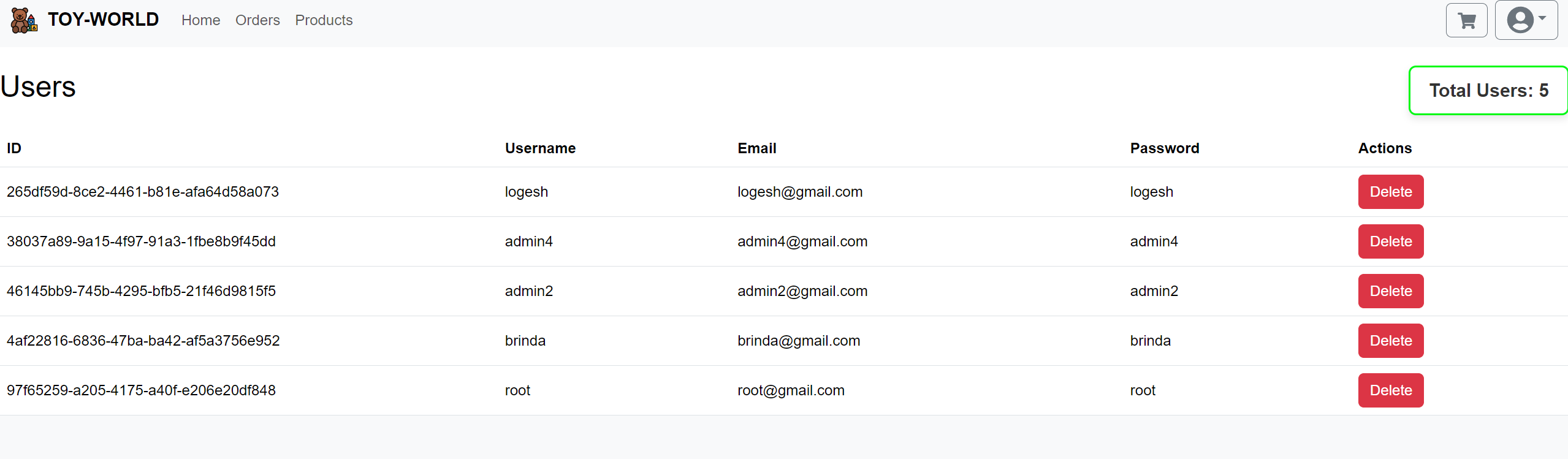
**ABSTRACT**

This report details the development of a toy store web application utilizing modern web technologies including React, Django, and SQL Workbench. The application follows the Model-View-Template (MVT) architecture, offering features such as user authentication, an admin dashboard, and product management

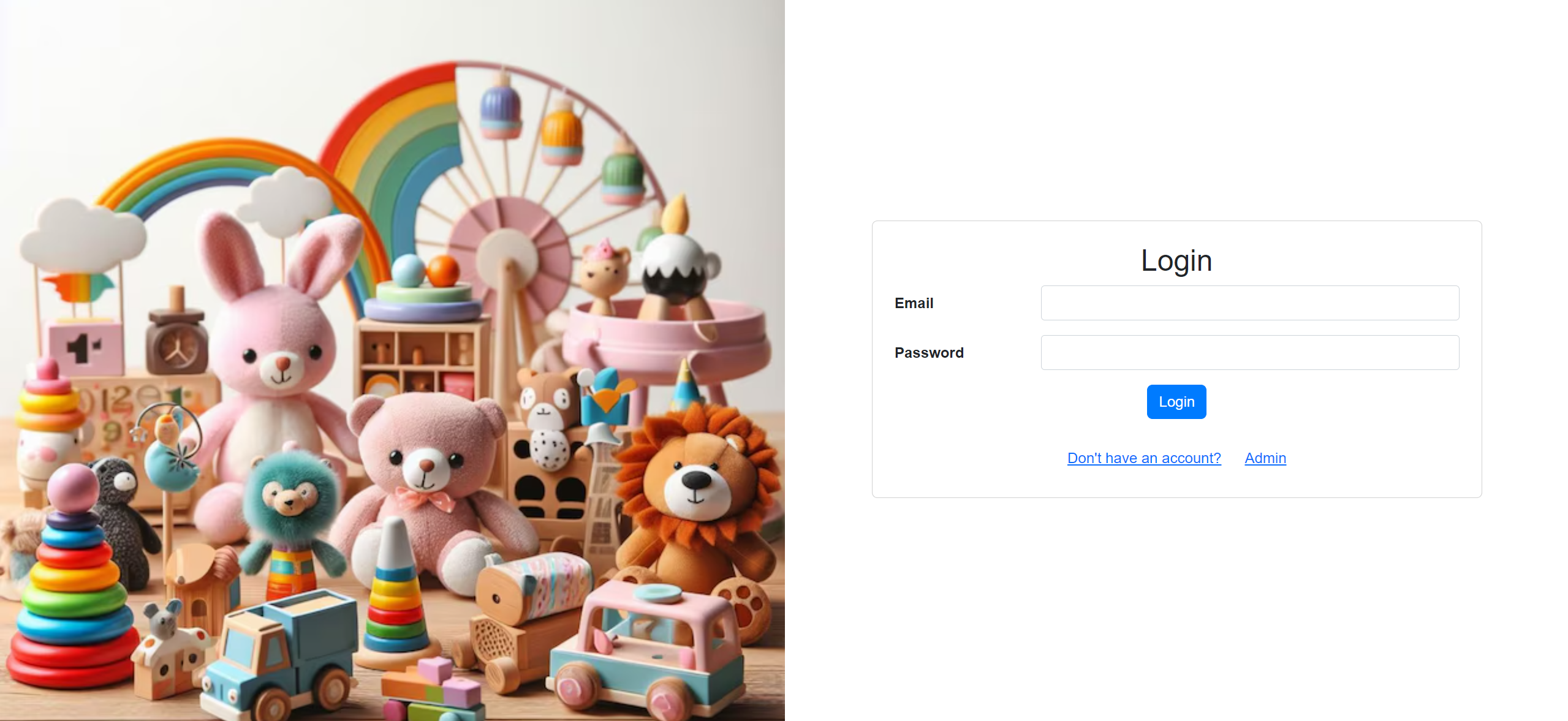
**LIST OF FIGURES**

**Project Architecture Overview**

****

**Admin Dashboard UI** 

User Signup Page UI



**CHAPTER 1: INTRODUCTION**

**1.1 Project Plan**

**1.1.1 About the Project**

The toy store app is designed to provide a seamless shopping experience for users, with functionalities for browsing, purchasing toys, and managing user accounts. The application also includes an admin dashboard for managing products and monitoring sales.

**1.1.2 Purpose and Scope**

The purpose of this project is to develop an e-commerce platform focused on toys, offering a user-friendly interface and efficient management tools for administrators. The scope includes user authentication, product catalog, shopping cart, and order management.

**1.2 Front End**

**1.2.1 Introduction of React**

React is utilized to build the dynamic and responsive user interface of the toy store app. Components such as the product list, shopping cart, and user authentication pages are all built using React.

**1.3 Back End**

**1.3.1 Python**

Python is the primary programming language used for the server-side of the application, with Django serving as the web framework.

**1.3.2 Django**

Django's Model-View-Template (MVT) architecture is implemented to manage the data flow and render the user interface. The framework is connected to SQL Workbench for database management.

**CHAPTER 2: SOFTWARE DEVELOPMENT LIFE CYCLE**

**2.1 Requirement Analysis Phase**

**2.1.1 System Requirement Specification**

**2.1.2 Hardware and Software Requirement**

Hardware: Standard development machine

Software: React, Django, SQL Workbench, Python, Node.js

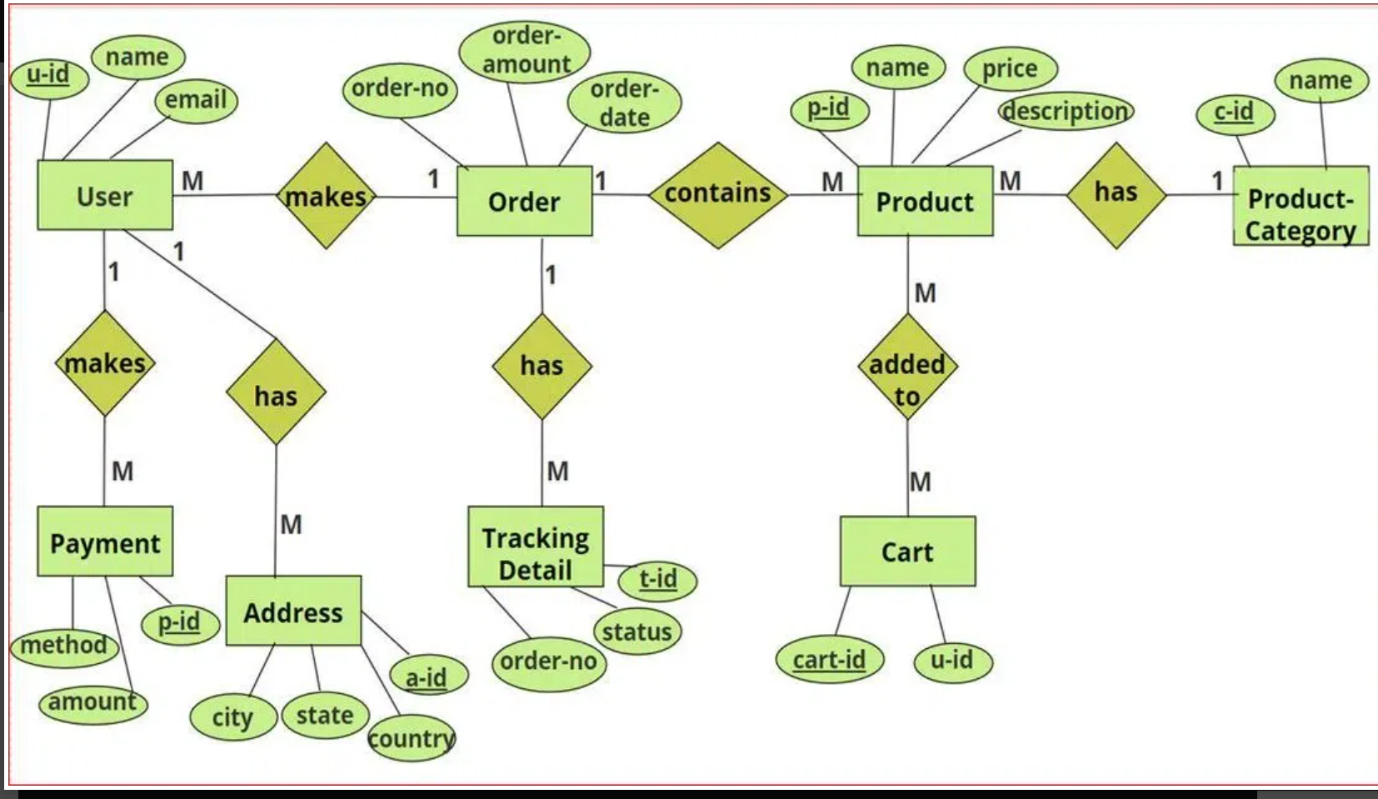
**2.1.3 Functional Requirement**

* User login and registration
* Admin dashboard for product management
* Product listing and shopping cart
* Order processing and checkout

2.1.4 Non-Functional Requirement

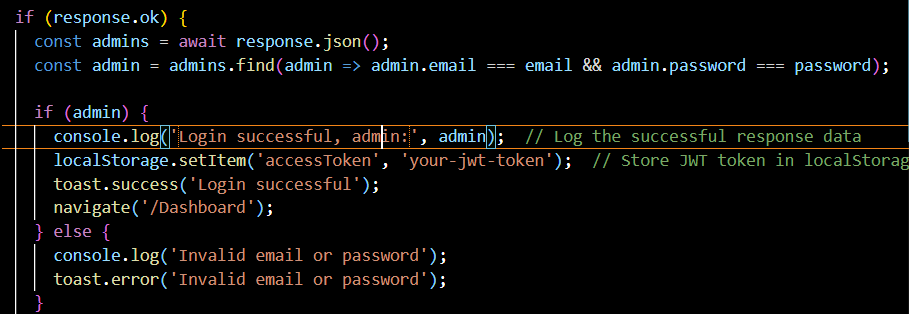
* Security: User data encryption and secure authentication
* Performance: Efficient loading times for the product catalog

CHAPTER 3: DIAGRAMS



CHAPTER 4: CODING

4.1 Python Code

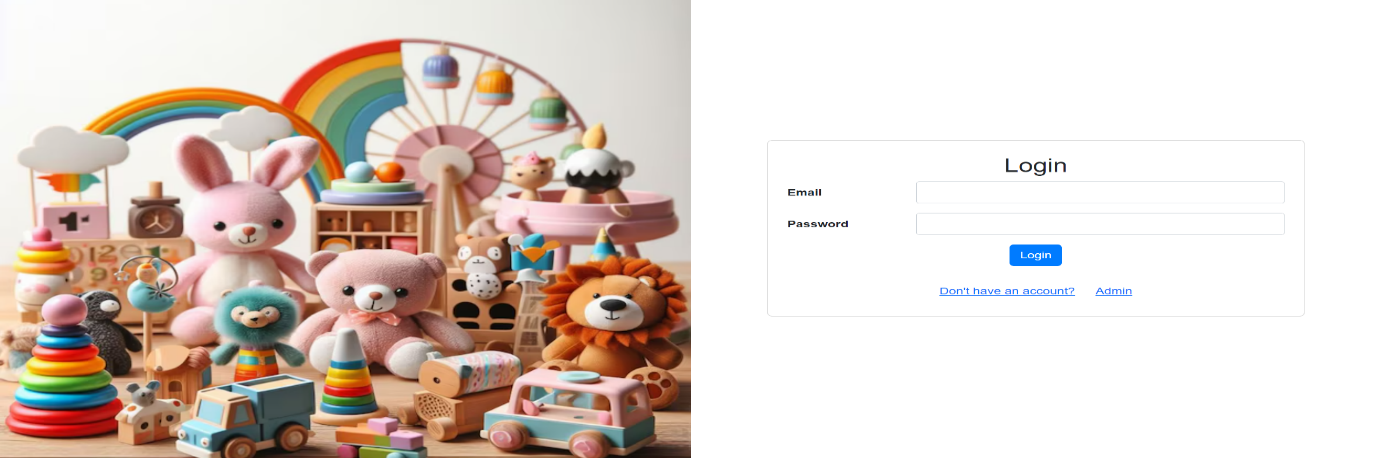


Example of product management logic

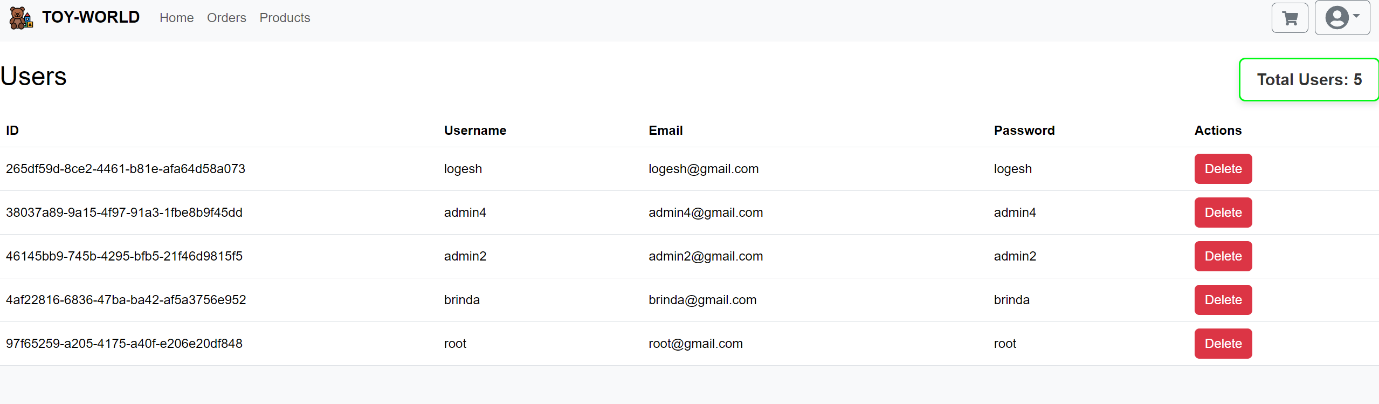


CHAPTER 5: SNAPSHOTS

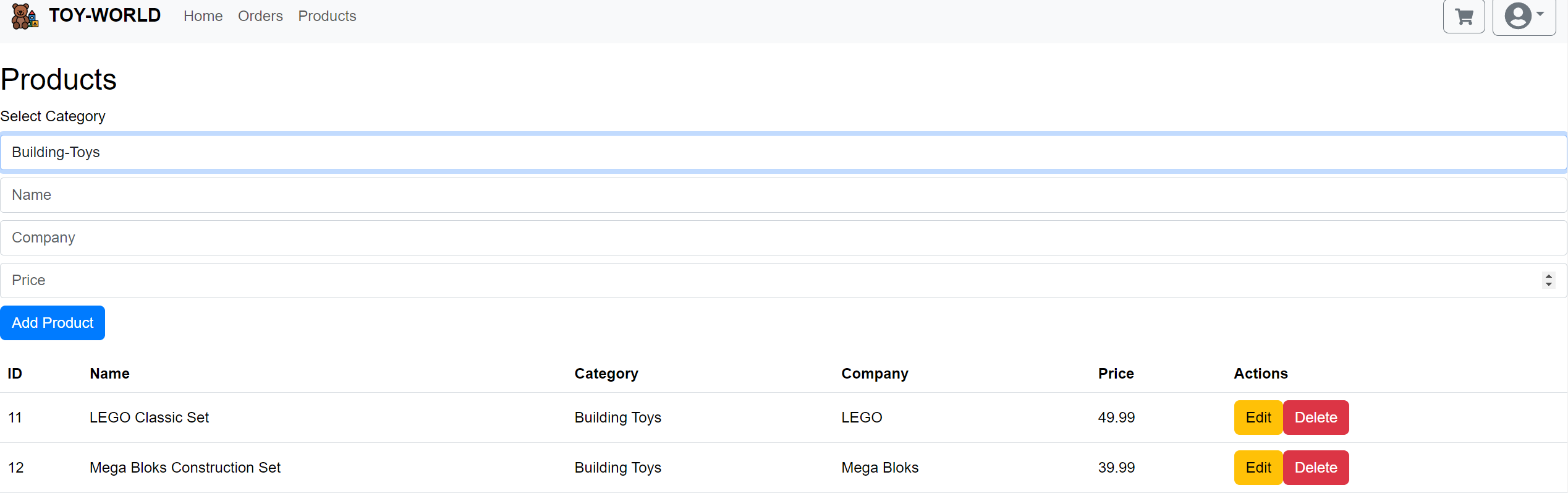
Login and Signup Page



Admin Dashboard

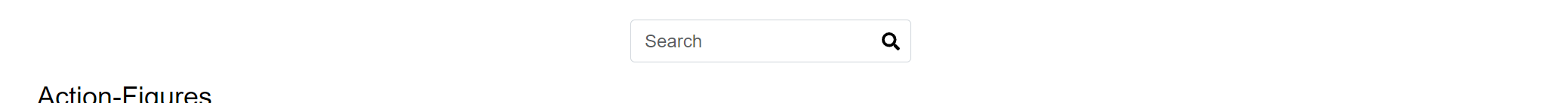


Product Catalog



CHAPTER 6: FUTURE ENHANCEMENT

Implementation of advanced search functionality



REFERENCES

Official React Documentation

Django Project Documentation

SQL Workbench User Guide

Python Programming Language Documentation