Executive Summary: BookStore DApp

> Zhenyuan Xiang, 1006739488

The BookStore DApp is a decentralized application designed to facilitate the buying, browsing, and discussion of books within a secure, blockchain-based environment, based on the seed DApp, Ethereum Pet Shop. It allows users to dynamically add book listings, view the available book collection, purchase books using cryptocurrency, and engage with other users by liking and commenting on their favorite titles. This application is built on the Ethereum blockchain, ensuring transparency and security in all transactions and interactions.

Key Features and Functionality:

1. **Dynamic Book Listings**: Unlike the original DApp, which featured a static list of items defined in a JSON file, the BookStore DApp introduces a new `BookStore.sol` smart contract. This contract enables dynamic addition of book listings, streamlining the process for sellers to update their inventory.
2. **Purchase Mechanism**: The DApp introduces a purchasing function by assigning a price attribute to each book within the `BookStore.sol` smart contract. Sellers can set prices for their books, and users must pay the specified amount in ETH to complete a purchase. This addition allows for a real-world shopping experience in a decentralized context.
3. **Book Categorization and Browsing**: To improve user experience, the DApp supports categorization of books. Each book is assigned a category, and users can browse the book listings based on these categories. This feature, implemented through mappings in the `BookStore.sol` contract, simplifies the search process for users looking for specific genres or types of books.
4. **Like and Sort by Popularity**: To enhance user engagement, the DApp includes a "like" feature. Each book has a like count attribute, and users can like books they find interesting. Additionally, books can be sorted by their like counts (or price and ID), allowing users to easily find the most popular titles. This functionality is also handled within the `BookStore.sol` smart contract.
5. **Comment System**: A separate `BookComments.sol` smart contract was developed to implement a comment system. Users can submit comments on individual books and view comment threads, facilitating discussion and the sharing of opinions. This feature fosters a community aspect within the DApp, where users can exchange ideas and reviews.

Relevance:

The BookStore DApp is relevant and interesting as it brings the concept of a traditional online bookstore to a decentralized platform. It leverages the benefits of blockchain technology, such as transparency, security, and the elimination of intermediaries, to offer a unique shopping and social experience. The DApp’s enhancements, particularly the dynamic listing of books, purchasing functionality, and user interaction features, address the limitations of the original application and provide a more robust and user-friendly experience.

Additions and Changes:

1. **Dynamic Book Listings**: In the `BookStore.sol` smart contract, a `Struct Book` was defined to store information about each book. The book list is stored in a `Book[] public books` array, and a `function addBook` was created to allow the addition of new books. On the front end, a form was added to facilitate the addition of new books by users.
2. **Purchase Mechanism**: A price attribute was added to each book in the smart contract, and a `function buyBook` was defined to handle the purchase process. When a user makes a payment, the revenue is automatically transferred to a pre-defined recipient. On the front end, the price of each book is now displayed.
3. **Book Categorization and Browsing**: A `mapping(string => uint[]) public IdsByCategory` was introduced in the smart contract to enable retrieval of book IDs by category. This was used in `App.js` to convert IDs into detailed book information. On the front end, a dropdown menu was added to allow users to define the category of a book when adding it, and to select the desired book type when viewing the book list.
4. **Like and Sort by Popularity**: A like counter was added to each book in the smart contract, along with a `function likeBook` to handle likes. A sorting method was added in `App.js` to allow sorting the book list by likes, price, or ID. On the front end, a dropdown menu was added to allow users to select the sorting method.
5. **Comment System**: A separate `BookComments.sol` smart contract was developed, where comments are stored in a `string[] public comments` array. A `mapping(uint => uint[]) private bookComments` was also defined to retrieve comment IDs based on the book ID. On the front end, a modal for comments was added to display the comment list and allow users to add new comments.

This DApp is a straightforward yet effective example of how traditional e-commerce functionality can be adapted to a decentralized environment, providing users with more control and privacy in their online transactions.