BetterReads

Kelsey, Ryan, Jarod

1. Welcome to the totally unique BetterReads, we have no affiliation with or relation to GoodReads. In all seriousness, we choose to make BetterReads a different website for reviewing and looking at books. Why? We like books and GoodReads got bought by Amazon, making Amazon more of a monopoly. If BetterReads were to be fully developed, and survive a copyright lawsuit, it could be a competitor to GoodReads in providing information about books. To make this project, Kelsey took up the frontend, Ryan took the backend, and Jarod served as a middle man. We started with selecting the data set to use within our selected domain of books. Our implementation of the project was disorganized at first due to our lack of knowledge in the matter. We did most of the checkpoints on time. We only really started moving a bit before the fourth checkpoint.
2. Implementation
   1. The system uses php for backend and html/JS for front end. The database used is the uww’s school database. Kelsey took up the frontend as mentioned above. She was able to make a nice website for our frontend. Ryan built most of the backend.
   2. We used Kaggle’s data on Amazon’s books and reviews. The data set is of books and their reviews. It has over 3 million rows with 20 columns. After getting it all, we split the data into multiple tables to organize the large data.
   3. ER diagram:
   4. Relational model:
      1. User(UserID, Email, Password, ProfilePic, Name)
         1. UserID is the primary key, Email and Password are also key
      2. Book(Title, Genre, Image, Price, Description, InfoLink, PreviewLink, Author)
         1. Title is the primary key
      3. Publisher(Name, PublishDate)
         1. Name is the primary key
      4. Reviews(ID, Score, Price, RatingCount, Helpfulness, Title, Time)
         1. ID is the primary key
      5. WantsRead(Image, Title, UserID)
         1. Image is primary key
      6. HasRead(Image, Title, UserID)
         1. Image is primary key
   5. The prototype has a purple and white theme for the website. It is able to take and store data for books and reviews. It would also show set data.
   6. We tested our website by recalling objects from the database. Then we tested to see if it could influence the database by adding and removing objects from the set database. We double checked the database to make sure the modifications of it went through.
3. Conclusion: We learned a lot about database management thanks to this project. We also came to understand the frustrations of getting the backend to work with the database. The biggest lesson was to plan and research before jumping into a project like this.