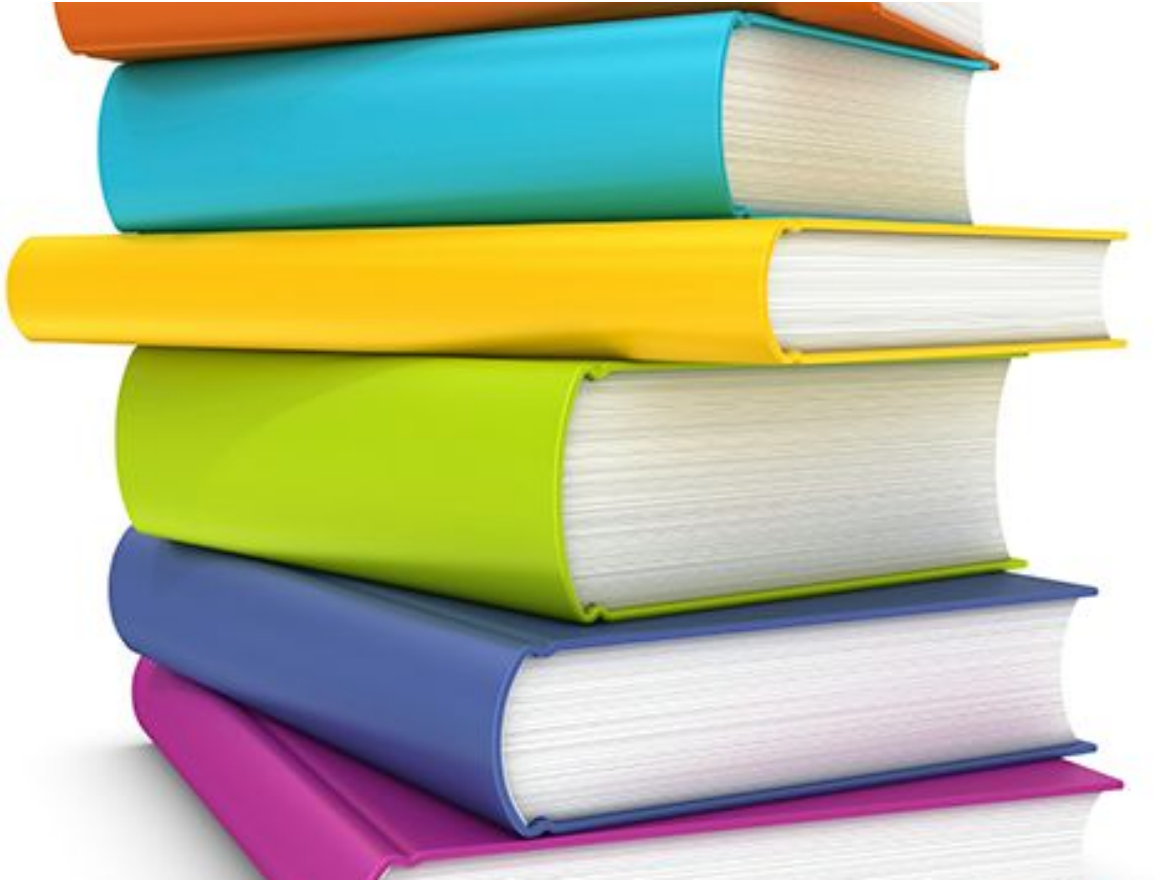


Bookeeper

Project 1: Report

<https://bookeeper-akashtyagi.c9.io/>



Akash

823008707

Database Systems

MOTIVATION

I am a huge fan of Mark Zuckerberg, and I was amazed when I came to know that he read 50 books a year. And he constantly post about his progress on Facebook. He motivated me to develop a habit for reading books. For that I wanted to create a website which will help me keep track of current books I am reading, already read and books for future reads. As a result, I came up with the idea of a personalized book management system for myself.

INTRODUCTION : BOOKEEPER

Bookeeper is a personalized book management system. It is a user friendly website which helps me in keeping track about status of my favourite books. Bookeeper has allowed me to track the progress of my books and enhanced my book reading experience.

My profile page gives a neat summary about the books already read, currently reading and other books available for reading, motivating me to read more books.

User Profile


<div>COVER PIC</div> 	Name	Email	Total Read	Currently Reading	Need To Read	Option
	akasht	akash.t@tamu.edu	7	3	1007	Update Profile

Fig 1. User Profile with summary

It allows me to add new books along with the information about the author and publisher. It automatically links the author and publisher, if pre-existing in database, otherwise create authors and publishers in the system.

Create Book

BOOK OPTIONS

[Profile](#)
[Already Read](#)
[Currently Reading](#)
[Books To Read](#)
[Search Book](#)
[Create New Book](#)

Title

Author

Publishing Year

Publisher

ISBN

Cover Image Link (jpg url)

Status

Unread ▼

Create

Fig 2. Create Book, Author and Publisher page

Author and Publishers profile pages gives information about all the published books in the system. This makes it easy to find other books by my favorite authors and publishers.

Publishers Profile Page

BOOK OPTIONS

[Profile](#)
[Already Read](#)
[Currently Reading](#)
[Books To Read](#)
[Search Book](#)
[Create New Book](#)

Id		Name		
220		Broadway		

Books Published

Book ID	Name	ISBN	Published	Option
821	The Sorcerer'S Companion: A Guide To The Magical World Of Harry Potter	0767908473	2001	Details
663	Passing For Thin : Losing Half My Weight And Finding My Self	0767912918	2004	Details

Fig 3. Publisher Profile Page

Author Profile Page

BOOK OPTIONS

[Profile](#)

[Already Read](#)

[Currently Reading](#)

[Books To Read](#)

[Search Book](#)

[Create New Book](#)

Id	Name
380	J.R.R. Tolkien

Books Published

Book ID	Name	ISBN	Published	Option
1171	The Book Of Lost Tales 2 (The History Of Middle-Earth - Volume 2)	034537522X	1992	Details
1181	The Hobbit	0618968636	2002	Details

Fig 4. Author Profile Page

Book profile page gives a neat summary about the book along with the status and cover pic of the book. Author and Publisher information is available on this page and by clicking the names of author or publisher, profile pages can be seen, hence able to see other books published by the same author or publisher.

Book Info Page

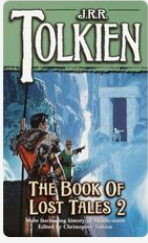
BOOK COVER PIC 	Title	Author	Publishing Year	Publisher	Status	Option
	The Book Of Lost Tales 2 (The History Of Middle-Earth - Volume 2)	J.R.R. Tolkien	1992	Del Rey	Unread	Update Book

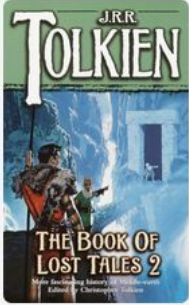
Fig 5. Book Profile Page

Update Book link available makes it extremely easy to update the information about the book, author and publisher. By changing the title, author name and publisher name updates the title and names respectively in the database. Thus, providing user a single stop for updating all the information about the book.

Book can be deleted too from the system but this does not delete the author and publisher from the system. Status of the book can also be changed from this page too.

Update Book Info

BOOK COVER PIC



BOOK OPTIONS

[Profile](#)
[Already Read](#)
[Currently Reading](#)
[Books To Read](#)
[Search Book](#)
[Create New Book](#)

Title

Delete Book

Delete

Author

ISBN

Publishing Year

Publisher

Status

Update

Fig 6. Book Update Page

Searching for a book is a really powerful and important tool available for a user. Finding a book among 1K tuples is really painful and time consuming. Searching makes it really easy to find book within seconds.

Search Book

BOOK OPTIONS

[Profile](#)
[Already Read](#)
[Currently Reading](#)
[Books To Read](#)
[Search Book](#)
[Create New Book](#)

Title

Author

Publishing Year/ Published Between

Publisher

Search

Fig 7. Book Search Page

Book can be searched on its title, author name, publisher name, publishing year and even on the range of the publishing year. Furthermore, combination of above can be used to narrow the search results for a particular book.

Another very important feature is to able to see the books read so far, currently reading and other books in the system. All the list are one click away from user making it really convenient to find the books of his choice.

Books Read So Far

BOOK OPTIONS Profile Already Read Currently Reading Books To Read Search Book Create New Book	Book ID	Name	ISBN	Published	Option
	18	The Testament	0440234743	1999	Details
	361	The Firm	0385416342	1991	Details
	513	Old Flames	0871138646	2003	Details
	592	The Partner	0385472951	1997	Details
	637	Burning Water (Burning Water)	0812524853	1992	Details
	821	The Sorcerer'S Companion: A Guide To The Magical World Of Harry Potter	0767908473	2001	Details
	1020	The Amazing Adventures Of Kavalier & Clay	0312282990	2001	Details

Fig 8. Books Read Page

DATA COLLECTION AND STORAGE

● Dataset

Data is collected from Institute of Informatic, University of Freiburg [1], from website <http://www2.informatik.uni-freiburg.de/~chiegler/BX/> . Dataset used is called Book-Crossing Dataset [2], which contains information about more than 1K books in sql format.

● Data Preprocessing

Data was preprocessed to break the single table of books into multiple tables. Specially for segregating the data of authors and publishers from single table. Main motivation to break single table into multiple table was to avoid data redundancy as much as possible.

Data preprocessing was done using Python scripts written by me and included in code.

- Storage

As suggested, MySQL is used for storing data and connected to the website using 'mysql' package. This is a node.js driver for mysql. It is written in JavaScript, does not require compiling, and is 100% MIT licensed.

DATABASE SCHEMA

The database schema is shown in figure 1.

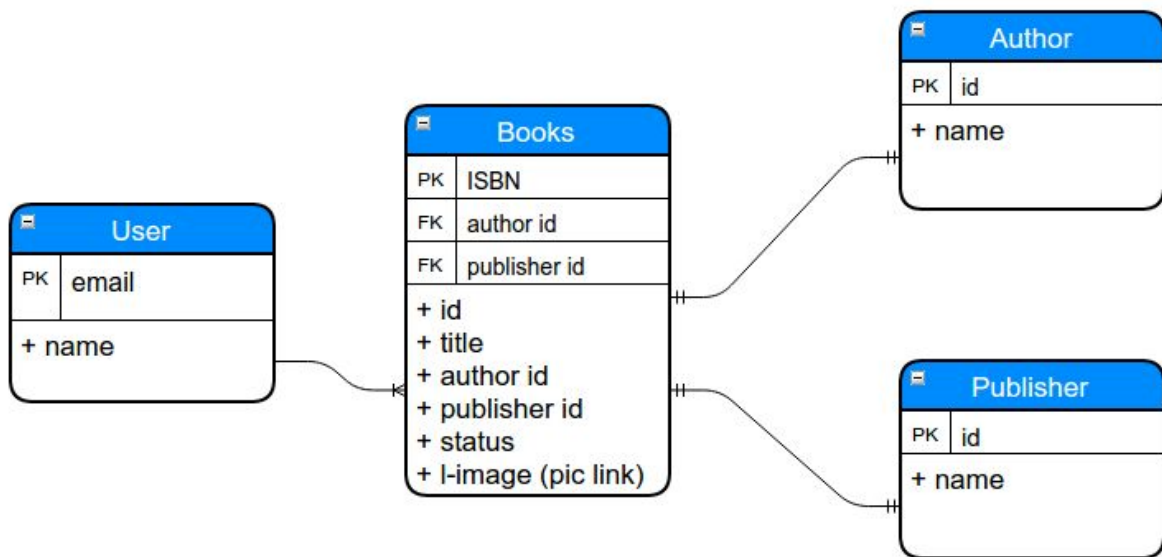


Fig 9. Database Schema for Bookeeper

I have 4 tables keeping track of my personal details, books, authors and publishers.

1. User: User table keeps track of the personal information of user using personalized website.

```
mysql> describe user;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name  | varchar(25)   | YES  |     | NULL    |       |
| email | varchar(100)  | NO   | PRI |         |       |
+-----+-----+-----+-----+-----+-----+
```

Fig 10. User table description

Various fields are:

- A. name: Used to store the name of the user.
 - B. email: Used to store the email id of the user and is the primary key for the table. It can not be null
2. Books: Books table stores the information about all the books in the system.

Field	Type	Null	Key	Default	Extra
ISBN	varchar(13)	NO	PRI		
id	int(11)	NO	UNI	NULL	auto_increment
title	varchar(255)	YES		NULL	
authorid	int(11)	YES	MUL	NULL	
published	int(10) unsigned	YES		NULL	
publisherid	int(11)	YES	MUL	NULL	
status	int(11)	YES		0	
linage	varchar(255)	YES		NULL	

Fig 11. Books table description

Various fields are:

- A. ISBN: ISBN stand for International Standard Book Identifier. It is a unique numeric book identifier which is used as primary key for the book table.
 - B. id: Unique identifier generated by automatically by the mysql database is used for giving an easy identifier to each book and is used extensively for querying the books table.
 - C. authorid: Unique id for an author is used which is a foreign key here and primary key in authors table.
 - D. published: The year on which the novel was published. Very useful in searching for novels of particular time era.
 - E. publisherid: Unique id for a publisher which is a foreign key here and primary key in publishers table.
 - F. status: Tells whether a user is currently reading, already read or is in the future read list.
 - G. limage: Short for Link for book's cover pic taken from www directly. Images instead of being stored in the system are available on cloud and are shown using link.
3. Authors: Stores the information about all the authors in the database.


```
mysql> describe authors;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50)   | YES  | UNI | NULL    |                |
+-----+-----+-----+-----+-----+-----+
```

Fig 12. User table description

Various fields are:

- A. id: Unique auto generated identifier is used as a primary key for the table.
 - B. name: Name of the author is considered unique for the table.
4. Publishers: Stores the information about all the publishers in the database.

```
mysql> describe publishers;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50)   | YES  | UNI | NULL    |                |
+-----+-----+-----+-----+-----+-----+
```

Fig 13. User table description

Various fields are:

- C. id: Unique auto generated identifier is used as a primary key for the table.
- D. name: Name of the publisher is considered unique for the table.

Table	Total Tuples
User	1
Books	1017
Authors	790

Publishers	399
------------	-----

Table 1. Number of tuples per table

PROGRAM COMPONENTS

1. Database: MySQL [3]

MySQL is an open-source relational database management system. MySQL is the world's most widely used (RDBMS) that runs as a server providing multi-user access to a number of databases.

2. Server: Node.js [4] (Express.js library [5])

Node.js is an open-source, cross-platform runtime environment for developing server-side web applications. Node.js provides an event-driven architecture and a non-blocking I/O API designed to optimize an application's throughput and scalability for real-time web applications.

Express.js is a Node.js web application server framework, designed for building single-page, multi-page, and hybrid web applications. It is the de facto standard server framework for node.js.

3. Front end: HTML, Jade [6] (Template Engine), Bootstrap [7] (CSS)

Jade is a high performance template engine heavily influenced by Haml and implemented with JavaScript for node and browsers.

Bootstrap is a free and open-source collection of tools for creating websites and web applications. Bootstrap is a front end framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server.

4. Repository Hosting Service: Github [8]

GitHub is a Web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features

5. Development Environment: Cloud 9 [10]

Cloud9 provides a development environment in the cloud that allows developers to get started with coding immediately and collaborate with their

peers.

DIFFICULTIES ENCOUNTERED AND MY SOLUTIONS:

1. Selection of development environment and programming components:

Given the power and popularity of Java and PHP, I was inclined to use them for the offline project development. But, when I started using the cloud based development environment and ease of web application using node.js, I was convinced that I should use **Cloud9 along with node.js, express.js and jade combination**. Cloud development made it possible to deploy application on web in just a minute, which also saved my time as I did not had to deploy website on department web server.

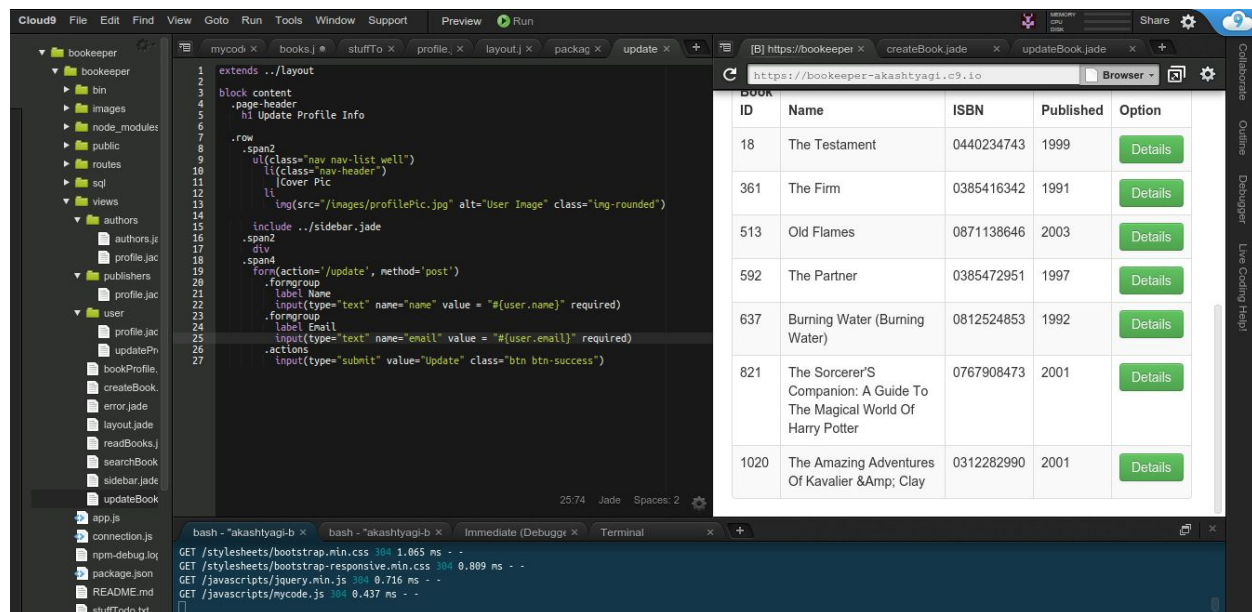


Fig 14. Cloud9 Development Environment

2. Show large data

Books table had more than 1K tuples which are all unread initially. Thus, showing the unread books become quite challenging as I couldn't dump all the 1K books to the user making the system slow and sometimes even crashing the web page. So, I decided to borrow the solution from **pagination** which is a process of dividing a document into multiple discrete pages.

Only 10 books at a time are shown to the user making it more easy for the user to decide what to read and much faster for the database to fetch books and faster for website to render the page. Using the next and prev links on the web page user can get next books or previous set of 10 books unread.

The rendering of the unread books webpage can be seen in figure 6.

Book ID	Name	ISBN	Published	Option
1	Classical Mythology	0195153448	2002	Details
2	Decision In Normandy	0060973129	1991	Details
3	Flu: The Story Of The Great Influenza Pandemic Of 1918 And The Search For The Virus That Caused It	0374157065	1999	Details
4	The Mummies Of Urumchi	0393045218	1999	Details
5	The Kitchen God'S Wife	0399135782	1991	Details
6	What If?: The World'S Foremost Military Historians Imagine What Might Have Been	0425176428	2000	Details
7	Pleading Guilty	0671870432	1993	Details
8	Under The Black Flag: The Romance And The Reality Of Life Among The Pirates	0679425608	1996	Details
9	Where You'LI Find Me: And Other Stories	074322678X	2002	Details
10	Nights Below Station Street	0771074670	1988	Details

[←Previous](#)
[→Next](#)

Fig 14. Pagination for unread books

3. Data Generation

Not only finding the right data for my idea was hard, but also cleaning it and breaking it into multiple tables was quite challenging too. For separating the authors and publishers data I had to write my own **python scripts**.

Other issue with the data was that some of the data was duplicated. Some authors and publishers had their names twice in the system. Since, I needed the name to be unique, I had to write script to remove the duplicate names from the data which was challenging too.

4. User friendly UI

I spent a lot of time in making UI look as user friendly as possible. All the nitty-gritty details of the backend database is hidden from the user making it a smooth process to use the website.

One feature which is really useful to the user is book, author and publishers creation web page. Since, the title of book, name of author and publisher is unique, the system is intelligent enough to realize when to create new book, author, publisher or link the book with already existing authors and publishers. Which avoids the hassle of separate author and publisher creation page.

Apart from that various options are available in the side widget which makes it really easy for user to access them.

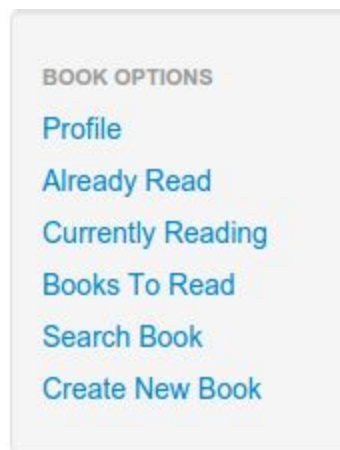


Fig 15. Book option widget

5. Schema generation

Coming up with a good schema is always challenging. Initially, I decided to store the book, author and publisher information in just one table books. But, soon I realized that there was too much data replication and since we were supposed to reduce the data redundancy as much as possible I decided to move the authors and publishers data in separate tables. But, having three tables made the database queries very challenging, but this is a tradeoff I am willing to have as this resulted in less data redundancy which is the goal of schema generation as mentioned in project description.

6. Updating multiple tables with single query

When creating a new book, a user can enter the name of new author as well as new

publisher. Thus, it became important to create all the three entries in the database in entirety. For this **transaction** came to my rescue which allows a sequential group of database manipulation operations to be executed as if it were one single unit.

7. Regular data corruption

During initial stages of development data corruption was rampant, wasting a lot of time to repopulate database. To speed up the process of data recovery I created sql scripts which when ran recovered data for me within seconds. **SQL scripts** were generated using python and saved a lot of effort on my part and made me more confident to develop web application more confidently.

LESSONS LEARNT FROM PROJECT:

Project gave me the picture on how to create a website and connect to database for persistent data. MySQL has proved really fast when data on the magnitude of thousands of tuples is used. During initial planning phase I decided to create indexes on various fields but realized that MySQL is really fast and there was no need to create indexes.

Using SQL commands like INSERT, UPDATE, CREATE, DELETE, JOINS and TRANSACTIONS have cleared my concepts and doubts regarding the working of these commands. Have made me more aware and careful while dealing with database, realized the importance of backup of data and ease of use of MySQL.

In the end, I feel I made something useful and worthwhile through this project which I am planning to continue to upgrade and use more often in my daily routine. This project has allowed me to create something which I can use daily and help me develop habit of book reading.

REFERENCES

1. <http://www.uni-freiburg.de/>
2. <http://www2.informatik.uni-freiburg.de/~ctiegle/BX/>
3. <https://www.mysql.com/>
4. <https://nodejs.org/en/about/>
5. <http://expressjs.com/>
6. <http://jade-lang.com/>

7. <http://getbootstrap.com/>
8. <https://github.com/about>
9. <https://github.com/felixge/node-mysql/>
10. <https://c9.io/site/about>
11. <https://en.wikipedia.org/wiki/MySQL>