

MARK MUINDE

SID 201374599

COMP2011 COURSEWORK 2

Introduction

For my coursework 2, I have chosen to implement a simple personalised record of books for users.

Here is the link to the website: <https://markmuinde.pythonanywhere.com>

The username is :markmuinde

The password is :password

1)Forms

In my forms.py file I have implemented a text field for the book title, a text area field for the book description (author, publisher; etc.) and the date they started reading the book as shown below:

#creating the form classes

```
class BookForm(Form):
    title = TextField('title', validators=[DataRequired()])
    description = TextAreaField('description', validators=[DataRequired()])
    date = DateField('date', format='%Y-%m-%d', validators=[DataRequired()])
```

2)Database

I have implemented 2 models, a booklist model that lists the book information and an author that has author info. An author can have many books and many books can have one author hence a many to many relationship. Nevertheless, I was only able to implement it as a one-to-many relationship i.e., one author can have many books as shown below:

```
#create the database table
class BookList(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    booktitle = db.Column(db.String(500))
    description = db.Column(db.String(500))
    startdate = db.Column(db.DateTime('%Y-%m-%d'))
    complete = db.Column(db.Boolean)
    author_id = db.Column(db.Integer, db.ForeignKey('author.id'))

    def __repr__(self):
        return '{0}{1}{2}{3}{4}{5}'.format(self.id, self.booktitle, self.description, self.startdate, self.complete, self.author_id)

class Author(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(500), index=True)
    contact_number = db.Column(db.String(20))
    books = db.relationship(
        'BookList', backref='author', lazy='dynamic')

    def __repr__(self):
        return '{0}{1}{2}'.format(self.id, self.name, self.contact_number, self.books)
```

3)Cookies

I have used simple GDPR compliance opensource code from [Cookie Consent by InSites](#). That displays a simple pop up alerting the user that the website is using cookies.

4)Styling

Given that it is simply a book record, I used simple styling in a separate CSS3 stylesheet as most avid readers are concerned with functionality more than beauty. On the left column of the webpage is unfinished books, in the middle finished books, and on the right a scrollable division for adding books to the database. A button class has been used to add books.

The screenshot shows a web application for a booklist. At the top, there is a light gray navbar with 'Navbar' and 'Home' links. Below the navbar is a dark gray banner with the text 'Hello, welcome to your booklist' in a light gray font. Underneath the banner, the heading 'Let's See What You're Reading' is displayed. The main content area is divided into three sections. On the left, 'Unfinished Books' lists 'The Three Musketeers' by Alexandre Dumas, noting it started on 2020-01-01 00:00:00 and providing a 'Mark As Complete' link. In the center, 'Completed Books' lists 'How The Mighty Fall', 'Teeth of The Tiger', 'King of the Fall', and 'City of Fallen Angels'. On the right, the 'Add a new book:' section contains input fields for 'Title', 'Description', and 'Start Date (YYYY-MM-DD)', followed by an 'Add Book' button. At the bottom of the page, a light blue footer contains a cookie notice with a 'Learn more' link and a 'Got it!' button.

4)Logging

I have used logging such that a useful message appears when the book list is successfully launched, when a book is successfully added, and a book is successfully marked as finished.

```
2021-01-18 17:04:56 INFO      BookList Launch successful
127.0.0.1 - - [18/Jan/2021 17:04:56] "GET / HTTP/1.1" 200 -

2021-01-18 17:02:11 INFO      Book successfully added
127.0.0.1 - - [18/Jan/2021 17:02:11] "POST /createbook HTTP/1.1" 302 -
2021-01-18 17:02:11 INFO      127.0.0.1 - - [18/Jan/2021 17:02:11] "POST /createbook HTTP/1.1" 302 -

2021-01-18 17:01:28 INFO      Book successfully marked as complete
127.0.0.1 - - [18/Jan/2021 17:01:28] "GET /completedbooks/3 HTTP/1.1" 302 -
2021-01-18 17:01:28 INFO      127.0.0.1 - - [18/Jan/2021 17:01:28] "GET /completedbooks/3 HTTP/1.1" 302 -
```

5)Features

As a simple fully functional booklist was implemented, I used Bootstrap and jQuery to make the website responsive. Nevertheless, along the way I found it more convenient to have all the information on one page, but I still left the features on the website as evidence of their use:

Navbar Home

Hello, welcome to your booklist

Let's See What You're Reading

Unfinished Books

- The Three Musketeers :

Alexandre Dumas

started on 2020-01-01 00:00:00

Mark As Complete

Completed Books

- How The Mighty Fall
- Teeth of The Tiger
- King of the Fall
- City of Fallen Angels

Add a new book:

Title

Description

Start Date (YYYY-MM-DD)

Add Book

This website uses cookies to ensure you get the best experience on our website. [Learn more](#)

Got It!

6)Security

Cross-Site Request Forgery has been prevented by the use of form validators.

On deploying the website, I found it more appropriate to use login security for the whole website.

The username is :markmuinde

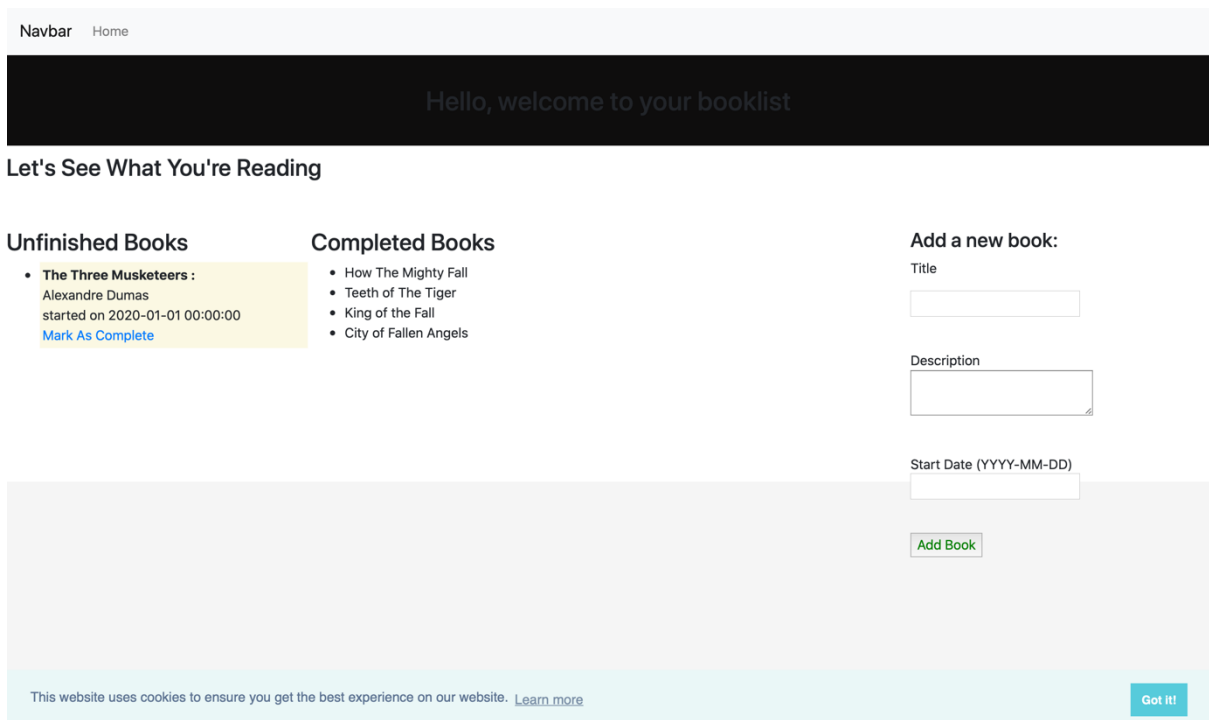
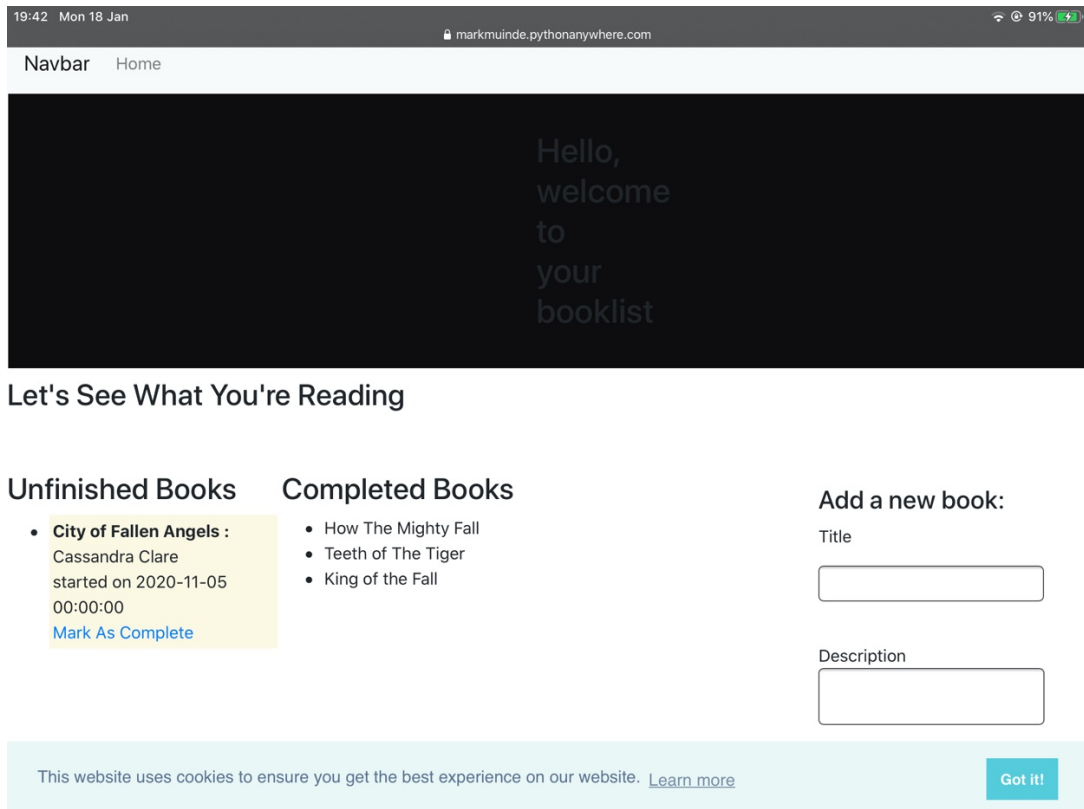
The password is :password

7)User Experience

I showed the application to 20 people all of whom agreed that for the purposes of a simple personal book record the website is satisfactory to a high standard as all the information they would want to document is easily documentable and the website itself is easy and intuitive to use.

8)Accessibility

The web app was built specifically for desktops and tablets and all data appears as required for both dimensions as shown on the next page. The first image is from an iPad and the second from a MacBook.



9) Deployment

The application has successfully been deployed on PythonAnywhere. The link can be found at the introduction.

10)References

a) Osano. 2021. *Download Cookie Consent | Osano*. [online] Available at:
<<https://www.osano.com/cookieconsent/download/>> [Accessed 18 January 2021].