Application Design

1. Registration Page:
    - URL : http://calm-plains-2686.herokuapp.com/register/
    - A new User can be created using a Registration Form
    - New user will be added to the list of Authenticated Users(AuthUser Model)

2. Login Page:
    - URL: http://calm-plains-2686.herokuapp.com/
    - User can login with an Existing username password
    - Directs it to the user's landing page with user specific information
    - This uses the AuthUser model

3. Landing Page:
    - URL: http://calm-plains-2686.herokuapp.com/landing/
    - Initially if the user is not logged in this will redirect to the login page.
    - Landing page contains following functionality:
        i. Shows the books reserved by the user sorted on reservation start time. Uses the Reservation Model
        ii. Total available books in the system sorted based on last reserved time. Uses the Books model filtered on the reservation.
        iii. Books owned by the user. Uses Books model.
        iv. Link to add a new Book
        v. Ability to delete an existing reservation
        vi. Clicking on the book name will route you to the book view page
        vii. Books which are not in the availability time will not be shown.

4. Show Book Page:
    - URL: (cannot be provided, depends on the selected user in landing page)
    - Shows the following information:
        i. Book name
        ii. Owner Name(linked to owner user page)
        iii. Reservation Information
        iv. Tags for the book(each tag linked to its page)
        v. RSS link(dumps the reservation information to XML format)
        vi. Link to Add Reservation(if user is not the owner)
        vii. Link to Edit the Book(if user is owner)
    - Validation:
        i. Only current and future reservations will be shown

        ii. Cannot add reservation if it's not between the availability time
        iii. While adding a reservation cannot provide negative duration
        iv. While editing resource cannot provide availability end time less than the start time.
    - Uses the Book Model
5. Show Owner Page:
    - URL: (cannot be provided, depends on the selected book in landing page)
    - Show the following information:
        i. Reservations made by the user
        ii. Books owned by the user
6. Tags Page:
    - URL : (cannot be provided, depends on the selected tags in landing page)
    - Shows the books which have the same tag
    - Uses the Tags and Tag_Book Model

7. Various Feature Information:
    - Page header has two links:
        i. The home icon will redirect you to the landing page
        ii. The Logout icon will logout the user and redirect to login page
    - All the date time fields take input using date picker widget, which is an add on and provided in the requirements.txt

Developer's Guide

- This application has been developed using python with Django framework. It is being hosted by Heroku server with the database being postgresql.
- It has been built with a basic MVC framework which is located inside the 'Hello' folder. Model.py is the model, veiws.py is the controller and templates are the views.
- The templates are written in html and CSS for the UI.
- Model.py contains the database structure which can be synced remotely to Postgresql server
- views.py contains the business logic.
- Django has a ModelForm utility which can be used to render the html input for forms. This functionality has been provided in forms.py.
- settings.py located in getting started contains provides us with the base dir root, database settings home page dir etc..

- urls.py takes care of routing the html to its respective view

Data Models

Apart from Django provided authentication tables I have created 4 new Models.

1. Book Model:
    a. Book_id: Primary Key, Auto Increment
    b. Owner_user: Foreign Key to User table;
    c. Avail_start: availability start time, DATETIME format
    d. Avail_end: availability end time, DATETIME format
    e. Name: Name of the book
    f. Last_reserve: automatically updated when a reservation is made
2. Reservation Model:
    a. Reserved_id: Primary Key, Auto Increment
    b. Book: foreign key to book table
    c. Reserved_user = Foreign key to user table. Input will be the id of the person making the reservation
    d. Reserved_Start: DATETIME field, mentions when the reservarion should start
    e. Reserved_end: DATETIME field, mentions when the reservarion should end
    f. Duration: duration of the reservation
3. Tag Model:
    a. Tag_id: primary key, auto increment
    b. Tag_name: name of the tag.
4. Tag_book: relational model between tag modeland book model
    a. Id : primary key, auto increment
    b. Tag: foreign key to the tag able
    c. Book: foreign key to book table