

Python/Django

Django

You may add this code to an existing project or create a new one, ideally we're testing your ability to add to an existing code base and your understanding of the Django MVC not how to setup a project.

Building a book

Level 1

Create a new django app called 'book'. The book app should have a model with at least the following classes: Book, Page, Section. We will have multiple Books and each book will have multiple pages and each page will have multiple sections. There are there types of book sections: Image, Text, and ImageText. We'll use the data model to display the book on the web so keep these things in mind:

- Pages and sections must have an order so they can be displayed correctly.
- There is no limit to the number of books in the system
- There is no limit to the number of pages in a book or the number of sections on a page.
- The Image section will contain image information, the Text section will contain text information and the ImageText section will contain both
- Use your knowledge to fill in the class attributes and functions, get creative but don't go overboard.

Level 2

We need to populate our book with some test data. Write an import function and adjust the model you created in level 1 so you can import the following json file.

```
{
  "books": [
    {
      "title": "This is my first book",
      "pages": [
        {
          "sections": [
            {
              "type": "text",
              "text": "This is the first section of page 1"
            },
            {
              "type": "text",
              "text": "This is the second section of page 1"
            },
            {
              "type": "image",
              "image": "http://lorempixel.com/400/200/"
            },
            {
              "type": "image_text",
              "text": "This is the first section of page 1",
              "image": "http://lorempixel.com/400/200/"
            }
          ]
        }
      ]
    }
  ],
  "sections": [
```

```

        {
            "type": "text",
            "text": "How did I get on page two...I'm lost."
        },
        {
            "type": "image",
            "image": "http://lorempixel.com/400/400/"
        },
        {
            "type": "text",
            "text": "This is how we do things down south of the
page..."
        }
    ],
    {
        "sections": [
            {
                "type": "text",
                "text": "I may be the last page but I'm the first
section."
            },
            {
                "type": "text",
                "text": "Why even try..why?"
            },
            {
                "type": "image_text",
                "text": "The end.",
                "image": "http://lorempixel.com/400/200/sports/1/"
            }
        ]
    }
]

```

```
}
    ]
}
```

Level 3

Using the book model you created above create views to browse the book. You should be able to browse a book by hitting a url `/book/ID/` where **I** **D** is the book id. If this is your first visit to this book you should start reading on page 1. If you've already browsed this book and left you should return to the page you left off from. The page should be rendered with each section shown in order. There should be two links on each page (Next/Prev) that progress the book forward and backwards. In between the two links you should add a textbox that allows the user to enter the exact page they want to go to with a submit button "Go" that takes them to that page.

Level 4

Damn, after all of this work we find out that some books have chapters and we need an index page for those books. The chapters are defined by any text page section that has an `<h1>` tag in it. Modify your code to add chapters to existing books and modify your import scripts to identify chapters upon import. For example, if a text section contains:

```
{
  "type": "text",
  "text": "<h1>Three little bigs</h1>"
},
```

You should identify this section as the beginning of a chapter and a new index page should appear at the front of the book that allows the user to instantly jump to this chapter. How you store this is up to you.

Level 5

My book is boring, I want some fun books on my website. Write a scraper as a Django management command that scrapes a short story from <http://www.eastoftheweb.com/short-stories/childrenindex.html> and inserts it into the database. The command should accept one argument (the story url).

Example urls:

- <http://www.eastoftheweb.com/short-stories/UBooks/Rapu.shtml>
- <http://www.eastoftheweb.com/short-stories/UBooks/FrogPrin.shtml>