LIBRARY MANAGEMENT SYSTEM

Arya Vinodan ,Abhishek Manoharan
Nithin M Warrier, Rahul Giridharan

College Of Engineering TVM

October 28, 2019

Overview

Introduction

With Advancements in Technology , it is now more desirable to update existing systems which uses manual addition of data into a database . For this we implement the following

- Using OCR to recognise characters and then adding to the Database
- Scanning the barcode getting url and adding to the Database.
- Both the methods minimises time taken to enter data and reduces human effort

Basically the goal of the project is to make the system less user dependent which would also drastically help in reducing induced costs and improves accessibility .

Problem Definition

The current system is not efficient and user friendly.We are trying to reduce the Manual effort in the normal library management system and to make management easy ,Ultimately reduce the cost.

1. Koha

koha

OPEN-SOURCE INTEGRATED LIBRARY SYSTEM

Koha is the first free software library automation package. In use worldwide, its development is steered by a growing community of users collaborating to achieve their technology goals

2. Book Cover recognition.

Linfeng Yang (linfeng@stanford.edu)
Xinyu Shen (xinyus@stanford.edu)

1. Image caption
2. Skew correction
3. MSER
4. Morphological filter
5. Positional filter
6. Bounding boxes merging
7. False detection inhibition and words auto correction

A. Image Preprocessing

- Image Capture
  - Loading an image in our database
  - Taking a real time image by webcam.

Skew Correction

The geometrical distortions caused by the digital camera or the rotations when people took the photo. Thus we need to correct the rotations problem.

MSER

we need to recognize which parts of the book contain text. MSER is a method that shows how we detect useful regions in an image that contains text. Basically, MSER will extract the regions where we observed large image gradient

Morphological filtering

Filter out non-text regions based on the geometric difference between text characters and non-text regions.

Positional filtering

Generally, the characters are well aligned in the text region on the book cover!!!!!,therefore the bounding boxes should be well aligned if they are real character boxes.

Bounding Boxes Merging

After the detection of individual character bounding boxes, we need to merge individual characters into words to perform better Optical Character Recognition.

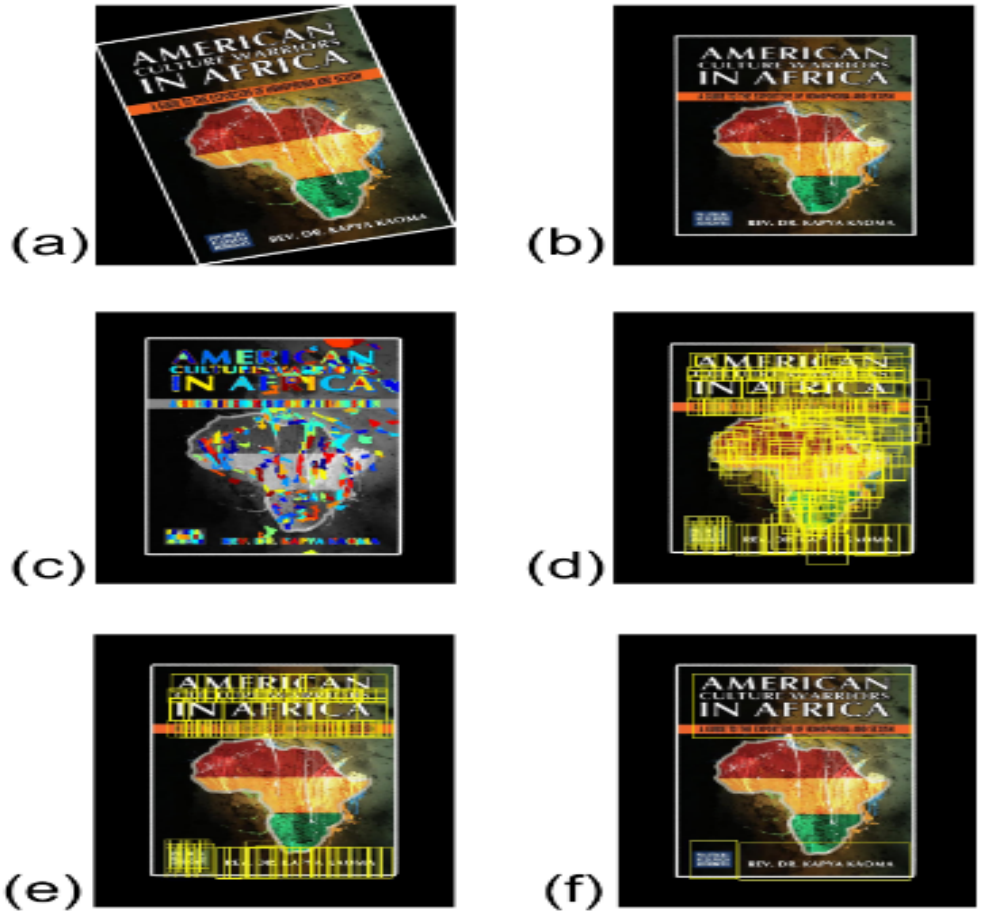


Fig. 2: Results for preprocess
(a): input image (b) de-skewed image (c) detect regions with MSER (d): find bounding box (e): remove non-aligned bounding box (f): merge bounding box

3. Web scraping

Pratiksha Ashiwal , S.R.Tandan , Priyanka Tripathi , Rohit Miri
https://www.ijraset.com/fileserve.php?FID=4999

Installation of BeautifulSoup

Beautiful Soup is a Python library for pulling data out of HTML and XML files.

Python Scripting

To extract data using web scraping with python, the following steps are to be followed:

1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required format

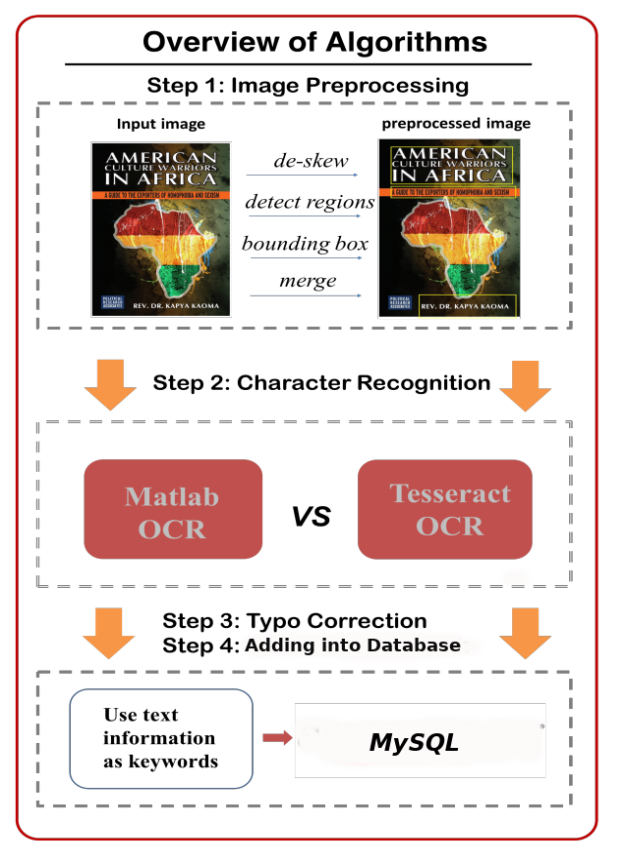
Store to DataBase

Data got from web scraping is stored into the required database.

Software Requirements

- OpenCV-Tesseract
- OpenCV-Pyzbar
- Python requests, BeautifulSoup
- MySQL

Problem Solution approach



Front cover recognition

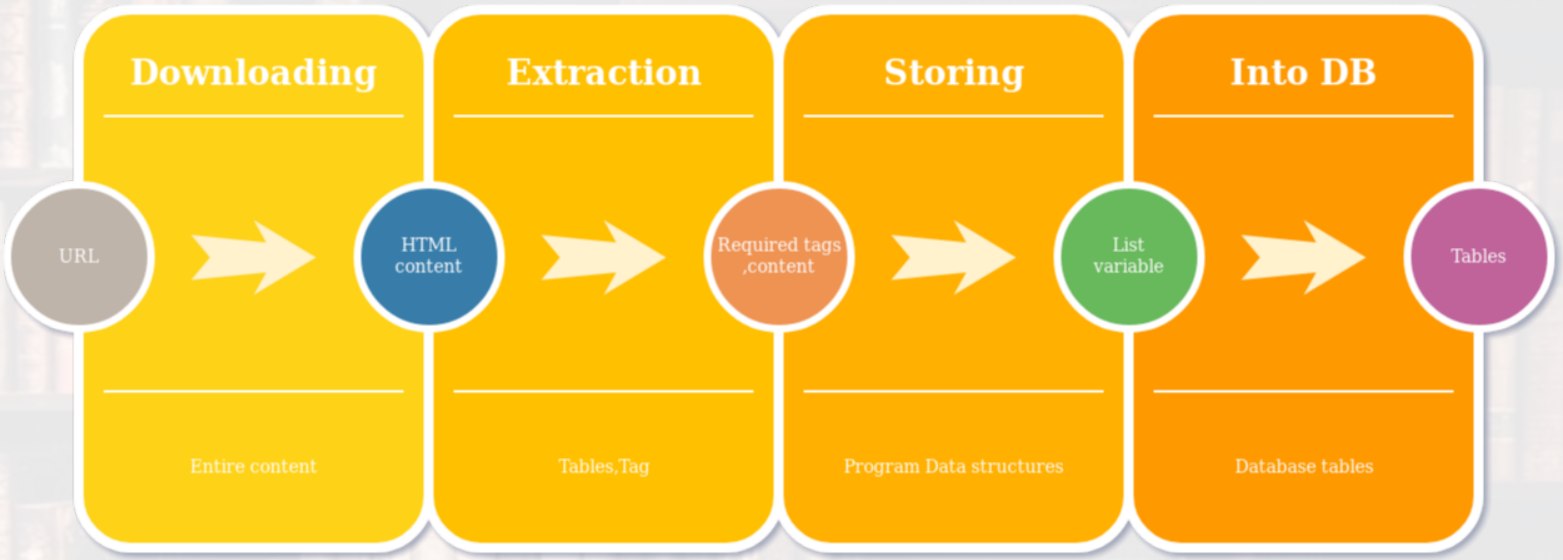
- Text recognition from front cover using Tesseract.
- Error Correction.
- Grouping the data accordingly.

Problem Solution approach

Bar code Scanning

- Scanning bar code using Opencv-Pyzbar.
- Receiving ISBN .
- Passing ISBN for web-scraping.

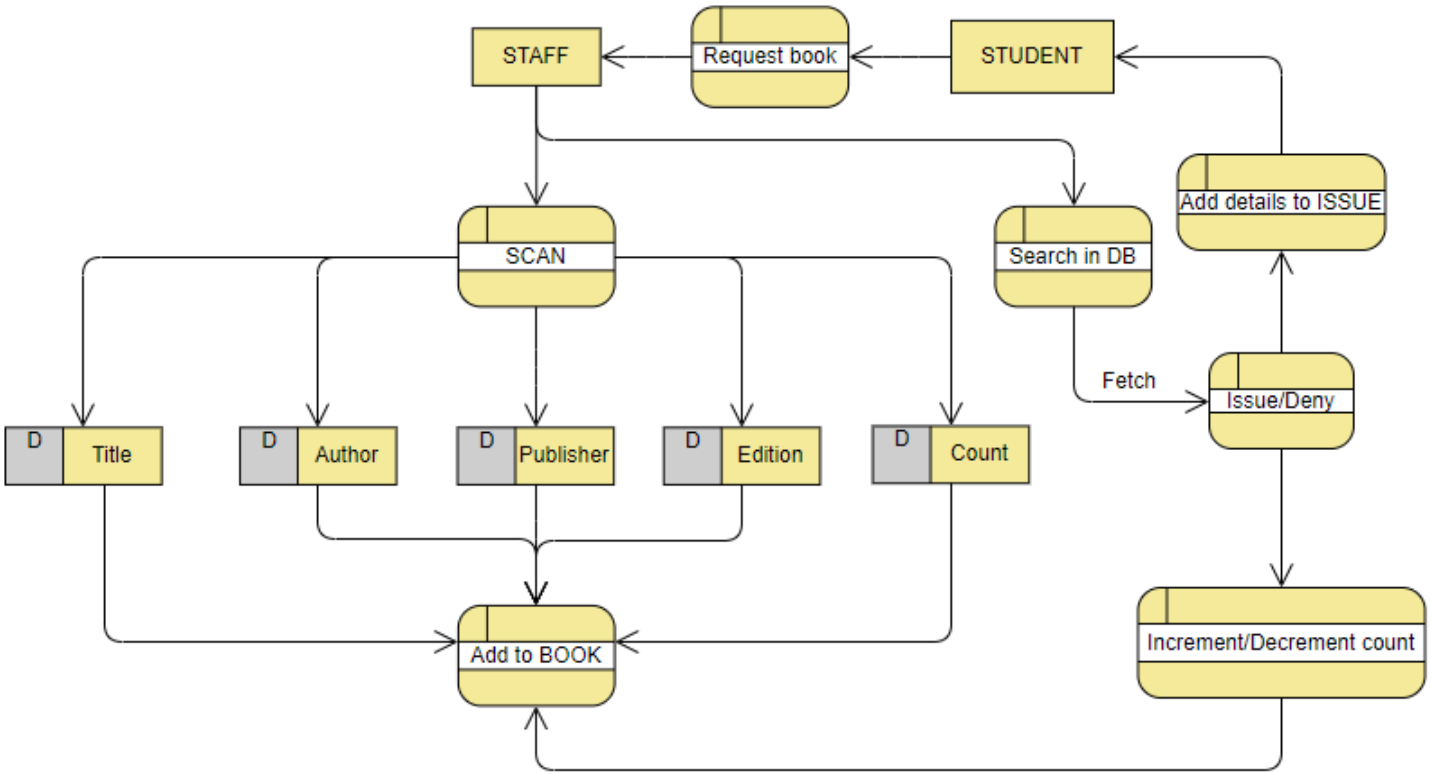
Problem Solution approach



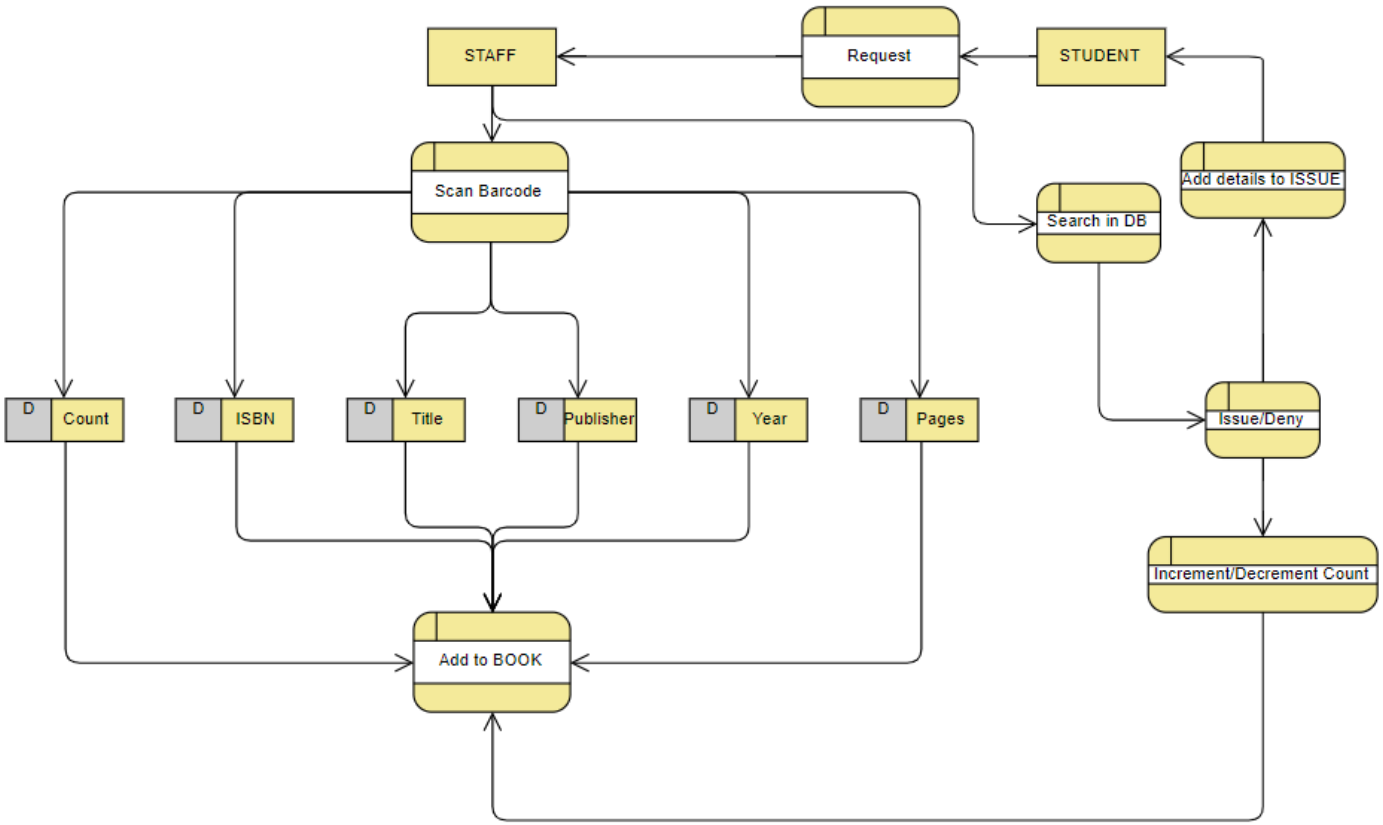
Web-Scraping

- Downloading content from web.
- Extracting content from table.
- Storing required data.
- Adding data into Database.

Data-Flow Diagram



Data-Flow Diagram



Current status of implementation

- Web scrapping with Python

home > abhishek > Documents > Design_project > URL.py > ...

```
1   import requests
2   from bs4 import BeautifulSoup as bs
3   import mysql.connector
4   from mysql.connector import Error
5   a=raw_input("Enter the URL: ")
6   webpage = requests.get(a)
7   webcontent = webpage.content
8   htmlcontent = bs(webcontent, 'html.parser')
9   prices=[] #values present
10  details = [] #what is the value
11  products = htmlcontent.find('table', {'id': 'metadata_content_table'})
12  for row in products.find_all('tr'):
13          for cells in row.find_all('td',{'class':'metadata_value'}):
14              prices.append(cells.text)
15          for cells in row.find_all('td',{'class':'metadata_label'}):
16              details.append(cells.text)
17  n = len(prices)
18  for i in range(n):
19      if details[i]=='Title':
20              t = i
21      elif details[i]=='Authors':
22              a = i
23      elif details[i]=='ISBN':
24              id = i
25      elif details[i] =='Publisher':
26              p = i
27      elif details[i] == 'Length':
28              l = i
29
30  name = prices[t]
31  isbn =prices[id]
32  part = isbn.split(",")
33  isbn =part[1]
34  isbn10 =part[0]
35  isbn =isbn.strip()
36  publisher =prices[p]
37  part = publisher.split(",")
38  publisher = part[0]
39  year = part[1]
```

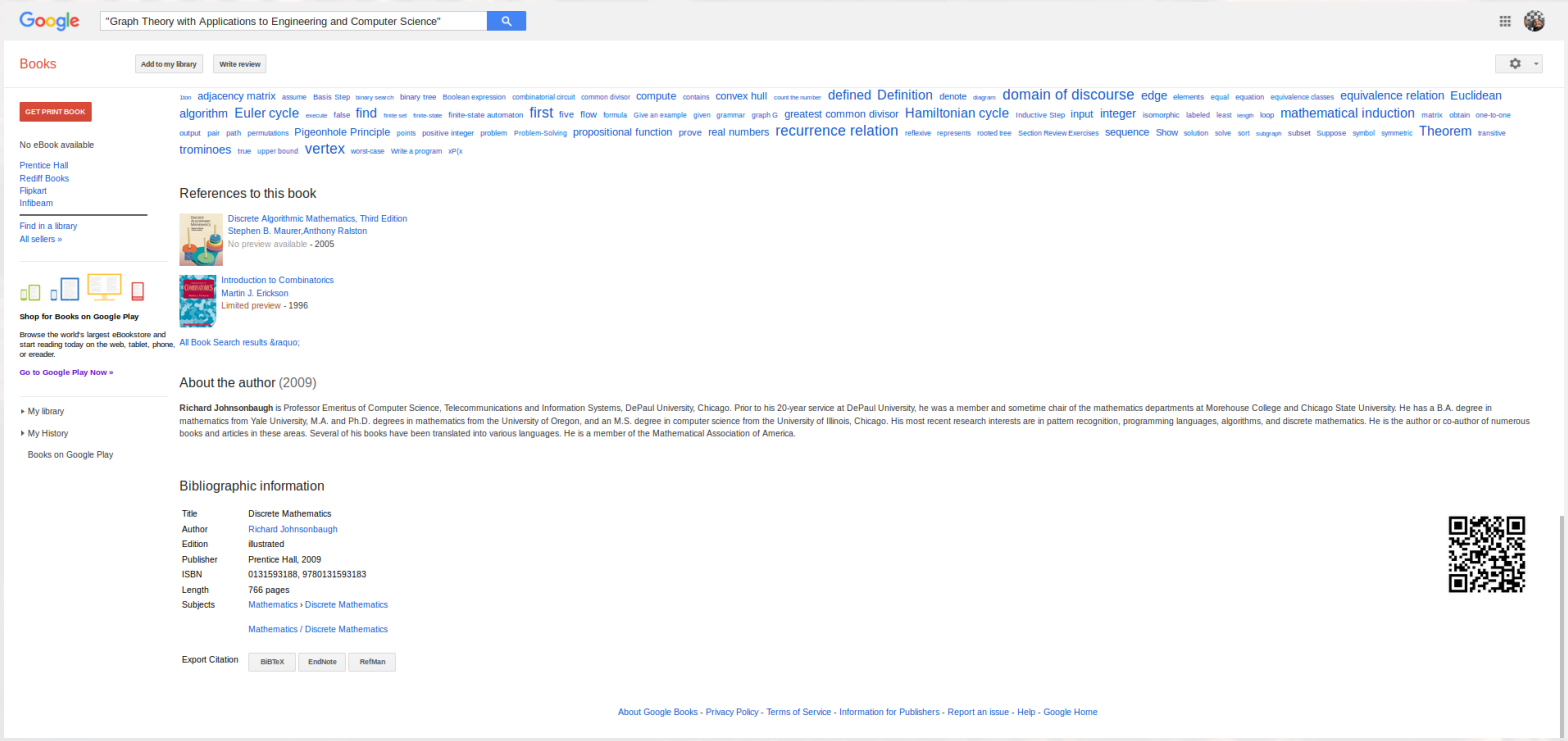
Current status of implementation

- Web scrapping with Python

```
#'''
try:
        connection = mysql.connector.connect(host='localhost',
                                             database='library',
                                             user='root',
                                             password='password')
        if connection.  connection ():
            db_Info = connection.get_server_info()
            print("Connected to MySQL Server version ", db_Info)
            cursor = connection.cursor()
            sql = """INSERT INTO Books( ISBN ,Title,publisher,year,pages ) VALUES (%s,%s,%s,%s,%s)"""
        #try:
            # Execute the SQL command
        recordTuple = ( isbn,name,publisher,year,pages )
        cursor.execute(sql,recordTuple)
        print "Data Successfully added"
            # Commit your changes in the database
        connection.commit()
        #except:
            #print "couldnt"
            # Rollback in case there is any error
            #connection.rollback()
except Error as e:
        print("Error while connecting to MySQL", e)
finally:
        if (connection.is_connected()):
            cursor.close()
            connection.close()
            print("MySQL connection is closed")
#'''
```

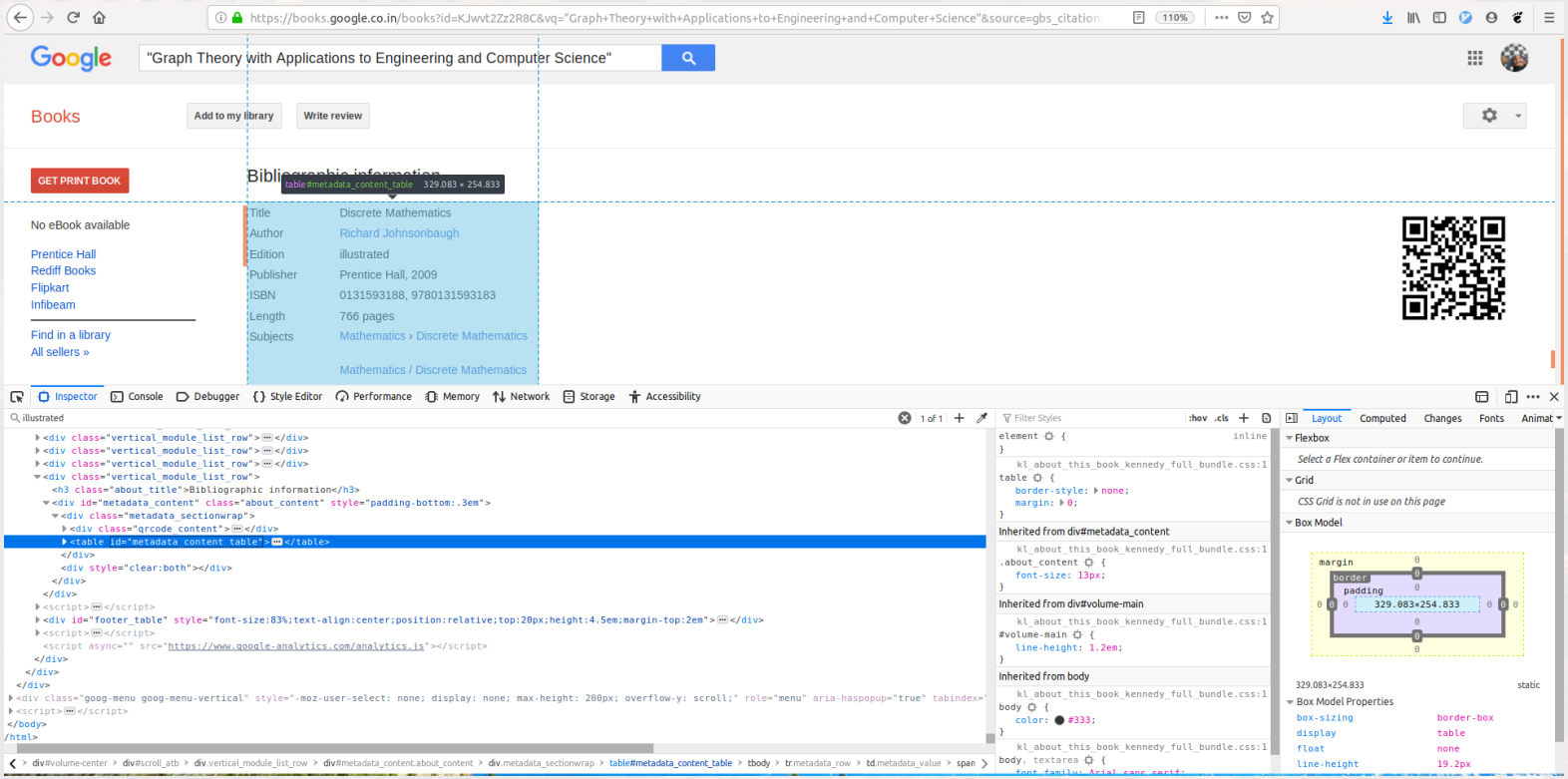
Current status of implementation

- Web scrapping with Python



Current status of implementation

- Web scrapping with Python



Current status of implementation

- Web scrapping with Python

```
abhishek@hephaestus:~/Documents/S1 & S2$ python webnewworking_userinput\ .py
Enter the URL: https://books.google.co.in/books?id=F9TIo4yuchoC&source=gbs_similarbooks
Discrete Optimization Algorithms: With Pascal ProgramsDover Books on Computer Science SeriesDover books on mathematics
Maciej M. Sys?o, Narsingh Deo, Janusz S. Kowalik
illustrated, reprint
Courier Corporation, 2006
0486453537, 9780486453538
542 pages
Computers › Programming › AlgorithmsComputers / Programming / AlgorithmsMathematics / Probability & Statistics / General
BiBTeX EndNote RefMan
abhishek@hephaestus:~/Documents/S1 & S2$
```

Current status of implementation

- Web scrapping with Python

```
abhishek@hephaestus:~/Documents/S1 & S2$ python URL.py
Enter the URL: https://books.google.co.in/books?id=KJwvt2Zz2R8C&vq=%22Graph+Theory+with+Applications+to+Engineering+and+Computer+Science%22&source=gbs_citatio
ns_module_r&cad=4
ISBN 9780131593183 name Discrete Mathematics publisher Prentice Hall year 2009 pages 766
('Connected to MySQL Server version ', u'5.7.27-0ubuntu0.18.04.1')
Data Successfully added
MySQL connection is closed
abhishek@hephaestus:~/Documents/S1 & S2$
```

Current status of implementation

- Web scrapping with Python

```
mysql> select * from library.Books;
+---------------+----------------------+---------------+------+-------+-------+
| ISBN          | Title                | Publisher     | year | pages | count |
+---------------+----------------------+---------------+------+-------+-------+
| 1234567890123 | dummy                | testing       | 2019 |   400 |     1 |
| 9780131593183 | Discrete Mathematics | Prentice Hall | 2009 |   766 |  NULL |
+---------------+----------------------+---------------+------+-------+-------+
2 rows in set (0.01 sec)

mysql>
```

Current status of implementation

- Bar-code Scanning

Barcode_scripts > barcode_scanner_image.py > ...

```python
# import the necessary packages
from pyzbar import pyzbar
import argparse
import cv2

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True,
    help="path to input image")
args = vars(ap.parse_args())

# load the input image
image = cv2.imread(args["image"])

# find the barcodes in the image and decode each of the barcodes
barcodes = pyzbar.decode(image)

# loop over the detected barcodes
for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the
    # bounding box surrounding the barcode on the image
    (x, y, w, h) = barcode.rect
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)

    # the barcode data is a bytes object so if we want to draw it on
    # our output image we need to convert it to a string first
    barcodeData = barcode.data.decode("utf-8")
    barcodeType = barcode.type

    # draw the barcode data and barcode type on the image
    text = "{} ({})".format(barcodeData, barcodeType)
    cv2.putText(image, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX
```

Current status of implementation

- Bar-code Scanning

```
abhishek@hephaestus:~/Documents/Design_project/Barcode_scripts$ python barcode_scanner_image.py --image  zbar-test1.jpg
[INFO] Found EAN13 barcode: 9781911223139
```

Current status of implementation

- Front cover scanning

```
abhishek@hephaestus:~/Documents/Design_project/images$ tesseract ss.jpeg stdout -l eng --oem 1 --psm 3
Warning. Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 282
System Software



BUNreBceltiot

Leland L. Beck = D. Manjula



PEARSON

abhishek@hephaestus:~/Documents/Design_project/images$
```

Current status of implementation

- Front cover scanning

```
abhishek@hephaestus:~/Documents/Design_project/images$ tesseract navathe.jpg stdout -l eng --oem 1 --psm 3
Warning. Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 1057


FUNDAMENTALS OF

DATABASE SYSTEM
JI “f



RAMEZ ELMASRI

Uy Pearson SHAMKANT B. NAVATHE
```

Conclusion

- As time progresses the technology advances at a high rate. It is with this advancement that we are able to reduce the pressure put on humans to complete various tasks.
- We believe that our project has many applications in the present and in the future.
- The project we designed when implemented will be of great use in the long run. Implementation in the college will provide us the first exposure to the project and can help us further polish it once the first test has been run thus aiding us in updating the project as much as possible

References

- Linfeng Yang(linfeng@stanford.edu),Xinyu Shen(xinyus@stanford.edu) *Book Cover Recognition*.
- Piyush Mankare, 2 Raman Kolekar. [*Advanced library management system*]. IJISET - International Journal of Innovative Science, Engineering Technology, Vol. 3 Issue 6, June 2016
- webscrapping
  `https://linuxhint.com/python-web-scraping-tutorial/`
  https://linuxconfig.org/introduction-to-python-web-scraping-and-the-beautiful-soup-library