**Elibrary with urls tutorials, pdf books & videos**

**INTRODUCTION:**

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter.

The aims and objectives are as follows:

• Online book reading.

• A search column to search availability of books.

• Facility to download required book.

• Video tutorial for students.

• An Admin login page where admin can add books, videos or page sources

• Open link for Learning Websites

E-Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can add new books, videos and Page sources. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used. All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized. In this chapter, we will discuss and analyze about the developing process of Library Management System including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

* 1. **Objective of the project:**

All existing library system where managing using manual entries or computer entries where admin or librarian has to record all details such as Book Name, student name, issue-date and applying penalty and all completing all this task is time consuming process and to overcome from this issue we are designing E-Library where ADMIN will upload various types of tutorials like Videos, tutorial URL and all types of files like PDF, WORD, PPT etc. Any student can signup with the application and then login and can search desired book with desired type. No manual management required

**2. LITERATURE SURVEY:**

**“Construction of a Digital Video Library: A Socio-Technical Pilot Study on College Students”**

This study investigates socio-technical aspects of digital video libraries based on college students' learning experiences and perspectives. Forty-one students in biology classes were studied through a survey and individual interviews. Findings are presented by the students' knowledge of computer technology, experiences with AV materials, and expectations of online AV materials.

**“Improving Browsing in Digital Libraries with Key phrase Indexes**”

Browsing accounts for much of people's interaction with digital libraries, but it is poorly supported by standard search engines. Conventional systems often operate at the wrong level, indexing words when people think in terms of topics, and returning documents when people want a broader view. As a result, users cannot easily determine what is in a collection, how well a particular topic is covered, or what kinds of queries will provide useful results. We have built a new kind of search engine, Key hind, that is explicitly designed to support browsing. Automatically extracted key phrases form the basic unit of both indexing and presentation, allowing users to interact with the collection at the level of topics and subjects rather than words and documents. The key phrase index also provides a simple mechanism for [clustering documents](https://www.sciencedirect.com/topics/computer-science/document-clustering), refining queries, and previewing results. We compared Keyphind to a traditional query engine in a small [usability study](https://www.sciencedirect.com/topics/computer-science/usability-study). Users reported that certain kinds of browsing tasks were much easier with the new interface, indicating that a keyphrase index would be a useful supplement to existing search tools.

**“A Digital Library Project on a Shoestring”**

Supported by a modest grant and collaboration with local historical societies, the California State University, Northridge, University Library assembled and created a digital archive of 2400 images of photographs, manuscripts, maps, postcards and brochures chronicling the history of the San Fernando Valley, a suburb of Los Angeles. In the process the library learned valuable lessons about administering a digital library project and its metadata. Supported by a modest grant and collaboration with local historical societies, the California State University, Northridge, University Library assembled and created a digital archive of 2400 images of photographs, manuscripts, maps, postcards and brochures chronicling the history of the San Fernando Valley, a suburb of Los Angeles. In the process the Library learned valuable lessons about administering a digital library project and its metadata. © 2002 Elsevier Science Ltd. All rights reserved.

**“Building A Digital Library: A Technology Manager’s Point of View”**

The Historic Pittsburgh project at the University of Pittsburgh can be instructive to others embarking upon digital library projects, and deciphering their own environmental opportunities and constraints.

**“From Information Gateway to Digital Library Management System: a case Analysis”**

This paper discusses the design, implementation and evolution of the Cornell University Library Gateway using the case analysis method. It diagnoses the Gateway within the conceptual framework of definitions and best practices associated with information gateways, portals, and emerging digital library management systems, in particular the product Encompass.

**“Imagining the Digital Library in a Commercialized Internet”**

Internet commerce is fueling technological innovation in marketing and customer relations, and transforming user expectations about Web sites that offer products and services. Digital library planning must address these trends in ways that effectively use changing technology, but also respect professional values of privacy, fairness, and disinterestedness.

**“Design and Implementation of Intelligent Library System”**

In this study, we combine swarm intelligence and Web Services to transform a conventional library system into an intelligent library system with high integrity, usability, correctness, and reliability software for readers. We select 300 readers to test this intelligent system and software, and compare it with a conventional library system. It is revealed that 64% of the readers are dissatisfied with the conventional library system, and 93% of the readers are satisfied with the intelligent library system when using personal digital assistants. The software integrity satisfaction was 99.99%; usability satisfaction, 93%; correctness, 95%; and reliability, 95%.

**“Digital Libraries: AN Overview”**

During the past recent years, there has been tremendous development reaming the concept of digital libraries-a knowledge base that can be stored and retrieved through on-line networks. Digital libraries are the most complex form of information systems that support digital document preservation, distributed database management, hypertext, filtering, information retrieval and selective dissemination of information. This has really overcome geographical barrier offering wide range of academic, research and cultural resources with multimedia effects which can be accessed around the world over the distributed networks. The paper examines the concept of Digital library, the technology that has enabled its emergence & architecture of digital library system. It also highlights the digital library projects undertaken in USA, UK and India. Here the authors explored the unique feature of digital library and possible challenges ahead for library and information professionals in the digital environment.

**“Beyond Information Searching and Browsing: Acquiring Knowledge from Digital Libraries”**

Digital libraries (DLs) are a resource for answering complex questions. Up to now, such systems mainly support keyword-based searching and browsing. The mapping from a research question to keywords and the assessment whether an article is relevant for a research question is completely with the user. In this paper, we present a two-layered digital library model. The aim is to enhance current DLs to support different levels of human cognitive acts, thus enabling new kinds of knowledge exchange among library users. The low layer of the model, namely, the tactical cognition support layer, provides users with requested relevant documents, as searching and browsing do. The upper layer of the model, namely, the strategic cognition support layer, not only provides users with relevant documents but also directly and intelligently answers users' cognitive questions. On the basis of the proposed model, we divide the DL information space into two subspaces, i.e., a knowledge subspace and a document subspace, where documents in the document subspace serves as the justification for the corresponding knowledge in the knowledge subspace. Detailed description of the knowledge subspace and its construction, as well as query facilities against the enhanced DLs for users' knowledge sharing and exchange, are particularly discussed

**3. SYSTEM ANALYSIS**

**Abstract:**

All existing library system where managing using manual entries or computer entries where admin or librarian has to record all details such as Book Name, student name, issue-date and applying penalty and all completing all this task is time consuming process and to overcome from this issue we are designing E-Library where ADMIN will upload various types of tutorials like Videos, tutorial URL and all types of files like PDF, WORD, PPT etc. Any student can signup with the application and then login and can search desired book with desired type. No manual management required

**Existing system:**

Digital library including video guides, downloadable PDF books, and useful website links for learning to date, libraries have been managed using a combination of manual and computerized processes that require a great deal of time and effort.

**Disadvantages:**

* More time taking

**Proposed system:**

In this project we used to know the library including video guides, downloadable PDF books, and useful website links for learning and this project we are using Django frame work and MySQL database. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a corporate network.

**Advantages:**

* Less time taking

**Modules Information:**

To implement this project, we have designed following modules

1. Admin Module: admin can login to application using username as ‘admin’ and password as ‘admin’ and then can add new book, video and URL etc. Admin can view all books and can delete book data.

2)Student Module: student can sign up and login to application and then can search desired type of tutorial by giving search query.

**FUNCTIONAL REQUIREMENTS:**

**SOFTWARE REQIREMENTS:**

**System Attributes:**

username,

password,

contact,

email,

address

**Data base Requirements:**

SQL

**USECASE:**

* Use cases - Use cases describe the interaction between the system and external users that leads to achieving particular goals.
* Each use case includes main elements:

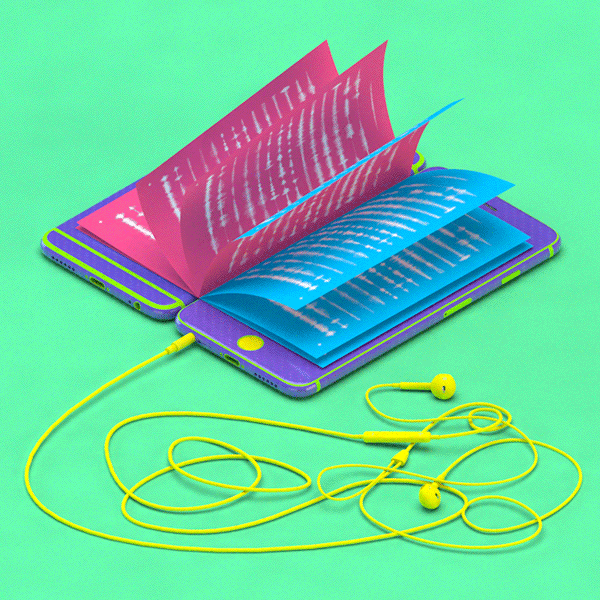
In this project we have designed two separate application such as admin module&student module.

1.admin module

2.student module

**User Stories:** we are designing E-Library where ADMIN will upload various types of tutorials like Videos, tutorial URL and all types of files like PDF, WORD, PPT etc.

**Work down Structure:**



**Prototype:**

python 3.7.0 or 3.7.4

opencv-python==4.5.1.48

keras==2.3.1

tensorflow==1.14.0

protobuf==3.16.0

h5py==2.10.0

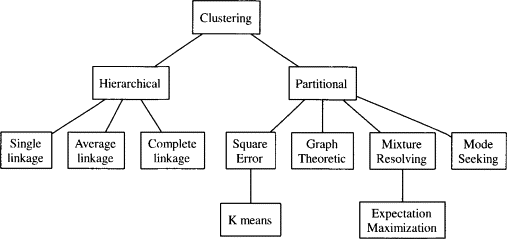
sklearn-extensions==0.0.2

scikit-learn==0.22.2.post1

Numpy

Pandas

**Models and Diagrams:**



**NON-FUNCTIONAL REQUIREMENT:**

**Usability:**  Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.(how it was handle entire project easy)

**Security:** the quality or state of being secure: such as. a : freedom from danger : safety. b : freedom from fear or anxiety. c : freedom from the prospect of being laid off job security.

**Readability:** Readability is the ease with which a reader can understand a written text.

**Performance**: the execution of an action. : something accomplished : deed, feat. : the fulfillment of a claim, promise, or request : implementation. 3. : the action of representing a character in a play.

**Availability**: the quality or state of being available trying to improve the availability of affordable housing. 2 : an available person or thing.

**Scalability**: Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

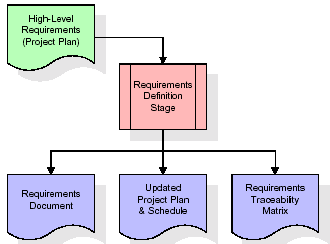
SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

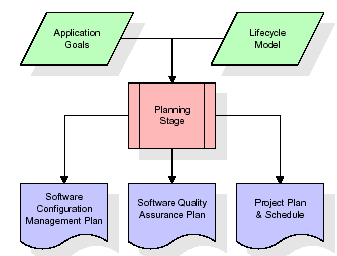
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

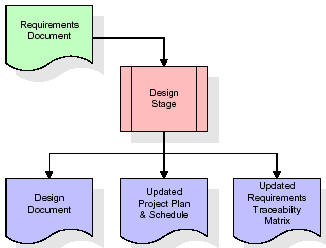
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

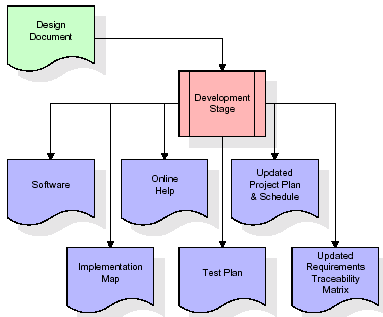
**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

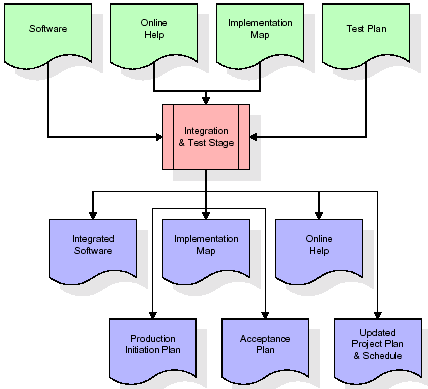
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

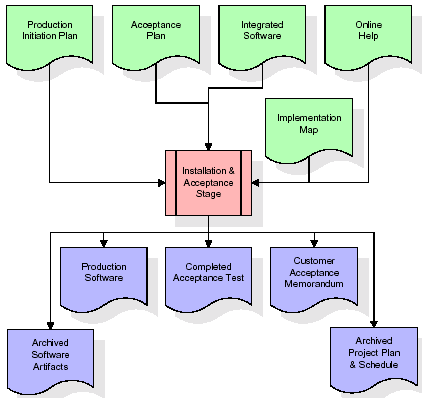


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loa ded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python

**4. SYSTEM DESIGN**

**4. SYSTEM DESIGN**

**CLASS DIAGRAM:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

****

**USECASE DIAGRAM:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



**SEQUENCE DIAGRAM:**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.





**COLLABORATION DIAGRAM:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.

****

****

**COMPONENT DIAGRAM:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**DEPLOYMENT DIAGRAM:**

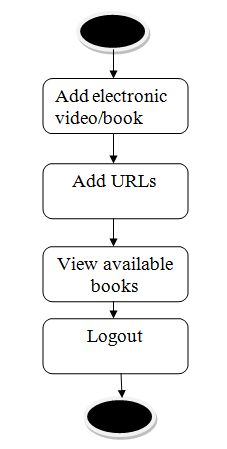
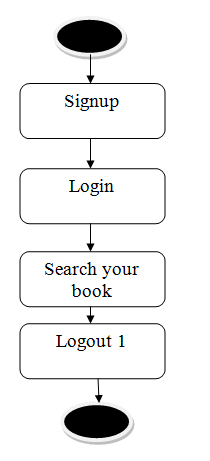
A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**ACTIVITY DIAGRAM:**

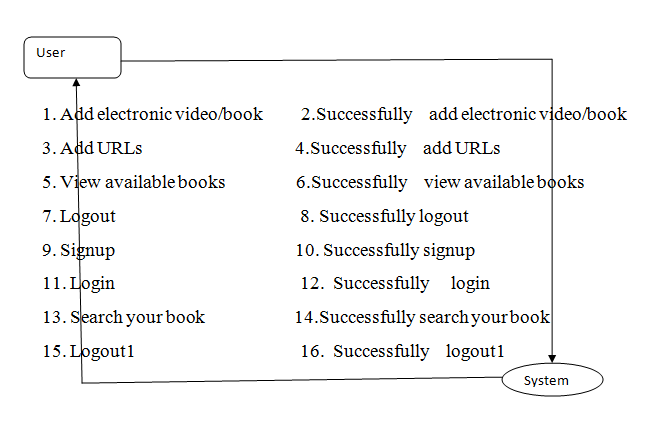
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.



**Data flow :**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

**from django.shortcuts import render**

**from django.template import RequestContext**

**from django.contrib import messages**

**from django.http import HttpResponse**

**from django.conf import settings**

**import os**

**import pymysql**

**from django.core.files.storage import FileSystemStorage**

**from datetime import date**

**global username, password, contact, email, address**

**def SearchBookAction(request):**

**if request.method == 'POST':**

**query = request.POST.get('t1', False)**

**file\_type = request.POST.get('t2', False)**

**query = query.lower()**

**array = query.split(" ")**

**output = '<table border=1 align=center width=100%>'**

**font = '<font size="" color="black">'**

**arr = ['ID','Name','Description','Book Date','Book Type','Filename','Access Data']**

**output += "<tr>"**

**dup = []**

**for i in range(len(arr)):**

**output += "<th>"+font+arr[i]+"</th>"**

**con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**with con:**

**cur = con.cursor()**

**cur.execute("select \* FROM addbook")**

**rows = cur.fetchall()**

**for row in rows:**

**book\_id = row[0]**

**book\_name = row[1]**

**description = row[2]**

**book\_date = row[3]**

**book\_type = row[4]**

**filename = row[5]**

**books = book\_name.lower()**

**descs = description**

**for k in range(len(array)):**

**if array[k] in books or array[k] in descs:**

**if filename not in dup and book\_type == file\_type:**

**dup.append(filename)**

**output += "<tr><td>"+font+str(book\_id)+"</td>"**

**output += "<td>"+font+book\_name+"</td>"**

**output += "<td>"+font+description+"</td>"**

**output += "<td>"+font+book\_date+"</td>"**

**output += "<td>"+font+book\_type+"</td>"**

**output += "<td>"+font+filename+"</td>"**

**if book\_type == "Video":**

**output+='<td><a href="PlayVideo?t1='+filename+'"><img src=/static/images/video.png height=100 width=100/></a></td>'**

**elif book\_type == "URL":**

**output+='<td><a href="'+book\_name+'" target="\_blank"><img src=/static/images/url.png height=100 width=100/></a></td>'**

**else:**

**output+='<td><a href="http://127.0.0.1:8000/static/books/'+filename+'"><img src=/static/images/book.jpg height=100 width=100/></a></td>'**

**context= {'data':output}**

**return render(request, 'SearchResult.html', context)**

**def SearchBook(request):**

**if request.method == 'GET':**

**return render(request, 'SearchBook.html', {})**

**def DeleteFile(request):**

**if request.method == 'GET':**

**filename = request.GET['t1']**

**db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**db\_cursor = db\_connection.cursor()**

**student\_sql\_query = "delete from addbook where file\_name = '"+filename+"'"**

**db\_cursor.execute(student\_sql\_query)**

**db\_connection.commit()**

**os.remove("LibraryApp/static/books/"+filename)**

**output = filename+' deleted from database'**

**context= {'data':output}**

**return render(request, 'AdminScreen.html', context)**

**def PlayVideo(request):**

**if request.method == 'GET':**

**video = request.GET['t1']**

**output = '<source src="static/books/'+video+'" type="video/mp4">Your browser does not support the video tag.'**

**context= {'data':output}**

**return render(request, 'PlayVideo.html', context)**

**def ViewBooks(request):**

**if request.method == 'GET':**

**output = '<table border=1 align=center width=100%>'**

**font = '<font size="" color="black">'**

**arr = ['ID','Name','Description','Book Date','Book Type','Filename','Access Data','Delete File']**

**output += "<tr>"**

**for i in range(len(arr)):**

**output += "<th>"+font+arr[i]+"</th>"**

**con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**with con:**

**cur = con.cursor()**

**cur.execute("select \* FROM addbook")**

**rows = cur.fetchall()**

**for row in rows:**

**book\_id = row[0]**

**book\_name = row[1]**

**description = row[2]**

**book\_date = row[3]**

**book\_type = row[4]**

**filename = row[5]**

**output += "<tr><td>"+font+str(book\_id)+"</td>"**

**output += "<td>"+font+book\_name+"</td>"**

**output += "<td>"+font+description+"</td>"**

**output += "<td>"+font+book\_date+"</td>"**

**output += "<td>"+font+book\_type+"</td>"**

**output += "<td>"+font+filename+"</td>"**

**if book\_type == "Video":**

**output+='<td><a href="PlayVideo?t1='+filename+'"><img src=/static/images/video.png height=100 width=100/></a></td>'**

**elif book\_type == "URL":**

**output+='<td><a href="'+book\_name+'" target="\_blank"><img src=/static/images/url.png height=100 width=100/></a></td>'**

**else:**

**output+='<td><a href="http://127.0.0.1:8000/static/books/'+filename+'"><img src=/static/images/book.jpg height=100 width=100/></a></td>'**

**output+='<td><a href="DeleteFile?t1='+filename+'">Click Here</a></td>'**

**context= {'data':output}**

**return render(request, 'ViewBooks.html', context)**

**def AddUrlAction(request):**

**if request.method == 'POST':**

**global username**

**url = request.POST.get('t1', False)**

**desc = request.POST.get('t2', False)**

**today = date.today()**

**output = "none"**

**count = 0**

**con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**with con:**

**cur = con.cursor()**

**cur.execute("select count(\*) from addbook")**

**rows = cur.fetchall()**

**for row in rows:**

**count = row[0]**

**count = count + 1**

**db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**db\_cursor = db\_connection.cursor()**

**student\_sql\_query = "INSERT INTO addbook(book\_id,book\_name,description,book\_date,book\_type,file\_name) VALUES('"+str(count)+"','"+url+"','"+desc+"','"+str(today)+"','URL','"+url+"')"**

**db\_cursor.execute(student\_sql\_query)**

**db\_connection.commit()**

**print(db\_cursor.rowcount, "Record Inserted")**

**if db\_cursor.rowcount == 1:**

**context= {'data':url+' Details saved in Database'}**

**return render(request, 'AddBook.html', context)**

**else:**

**context= {'data':'Error in adding book details'}**

**return render(request, 'AddBook.html', context)**

**def AddUrl(request):**

**if request.method == 'GET':**

**return render(request, 'AddUrl.html', {})**

**def AddBook(request):**

**if request.method == 'GET':**

**return render(request, 'AddBook.html', {})**

**def AddBookAction(request):**

**if request.method == 'POST':**

**global username, password, contact, email, address**

**name = request.POST.get('t1', False)**

**desc = request.POST.get('t2', False)**

**book\_type = request.POST.get('t4', False)**

**book\_name = request.FILES['t3'].name**

**book\_data = request.FILES['t3']**

**fs = FileSystemStorage()**

**today = date.today()**

**output = "none"**

**count = 0**

**con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**with con:**

**cur = con.cursor()**

**cur.execute("select count(\*) from addbook")**

**rows = cur.fetchall()**

**for row in rows:**

**count = row[0]**

**count = count + 1**

**db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**db\_cursor = db\_connection.cursor()**

**student\_sql\_query = "INSERT INTO addbook(book\_id,book\_name,description,book\_date,book\_type,file\_name) VALUES('"+str(count)+"','"+name+"','"+desc+"','"+str(today)+"','"+book\_type+"','"+book\_name+"')"**

**db\_cursor.execute(student\_sql\_query)**

**db\_connection.commit()**

**print(db\_cursor.rowcount, "Record Inserted")**

**if db\_cursor.rowcount == 1:**

**fs.save('LibraryApp/static/books/'+book\_name, book\_data)**

**context= {'data':book\_type+' Details saved in Database'}**

**return render(request, 'AddBook.html', context)**

**else:**

**context= {'data':'Error in adding book details'}**

**return render(request, 'AddBook.html', context)**

**def UserLogin(request):**

**global username**

**if request.method == 'POST':**

**username = request.POST.get('t1', False)**

**password = request.POST.get('t2', False)**

**con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**output = "none"**

**with con:**

**cur = con.cursor()**

**cur.execute("select username,password FROM register")**

**rows = cur.fetchall()**

**for row in rows:**

**if row[0] == username and row[1] == password:**

**username = row[0]**

**output = "success"**

**break**

**if output != 'none':**

**context= {'data':output}**

**return render(request, 'UserScreen.html', context)**

**if output == 'none':**

**context= {'data':'Invalid username'}**

**return render(request, 'Login.html', context)**

**def Register(request):**

**if request.method == 'GET':**

**return render(request, 'Register.html', {})**

**def index(request):**

**if request.method == 'GET':**

**return render(request, 'index.html', {})**

**def Login(request):**

**if request.method == 'GET':**

**return render(request, 'Login.html', {})**

**def AdminLogin(request):**

**if request.method == 'GET':**

**return render(request, 'AdminLogin.html', {})**

**def RegisterAction(request):**

**if request.method == 'POST':**

**global username, password, contact, email, address**

**username = request.POST.get('t1', False)**

**contact = request.POST.get('t3', False)**

**email = request.POST.get('t4', False)**

**address = request.POST.get('t5', False)**

**password = request.POST.get('t2', False)**

**fs = FileSystemStorage()**

**output = "none"**

**con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**with con:**

**cur = con.cursor()**

**cur.execute("select username,email FROM register")**

**rows = cur.fetchall()**

**for row in rows:**

**if row[0] == username:**

**output = username+" Username already exists"**

**break**

**if row[1] == email:**

**output = email+" Email id already exists"**

**break**

**if output == "none":**

**db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'elibrary',charset='utf8')**

**db\_cursor = db\_connection.cursor()**

**student\_sql\_query = "INSERT INTO register(username,password,contact,email,address) VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"')"**

**db\_cursor.execute(student\_sql\_query)**

**db\_connection.commit()**

**print(db\_cursor.rowcount, "Record Inserted")**

**if db\_cursor.rowcount == 1:**

**context= {'data':'Signup Process Completed'}**

**return render(request, 'Register.html', context)**

**else:**

**context= {'data':'Error in signup process'}**

**return render(request, 'Register.html', context)**

**else:**

**context= {'data':output}**

**return render(request, 'Register.html', context)**

**def AdminLoginAction(request):**

**if request.method == 'POST':**

**user = request.POST.get('t1', False)**

**password = request.POST.get('t2', False)**

**if user == 'admin' and password == 'admin':**

**context= {'data':'Welcome '+user}**

**return render(request, 'AdminScreen.html', context)**

**else:**

**context= {'data':'Invalid login'}**

**return render(request, 'AdminLogin.html', context)**

**6. TESTING:**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

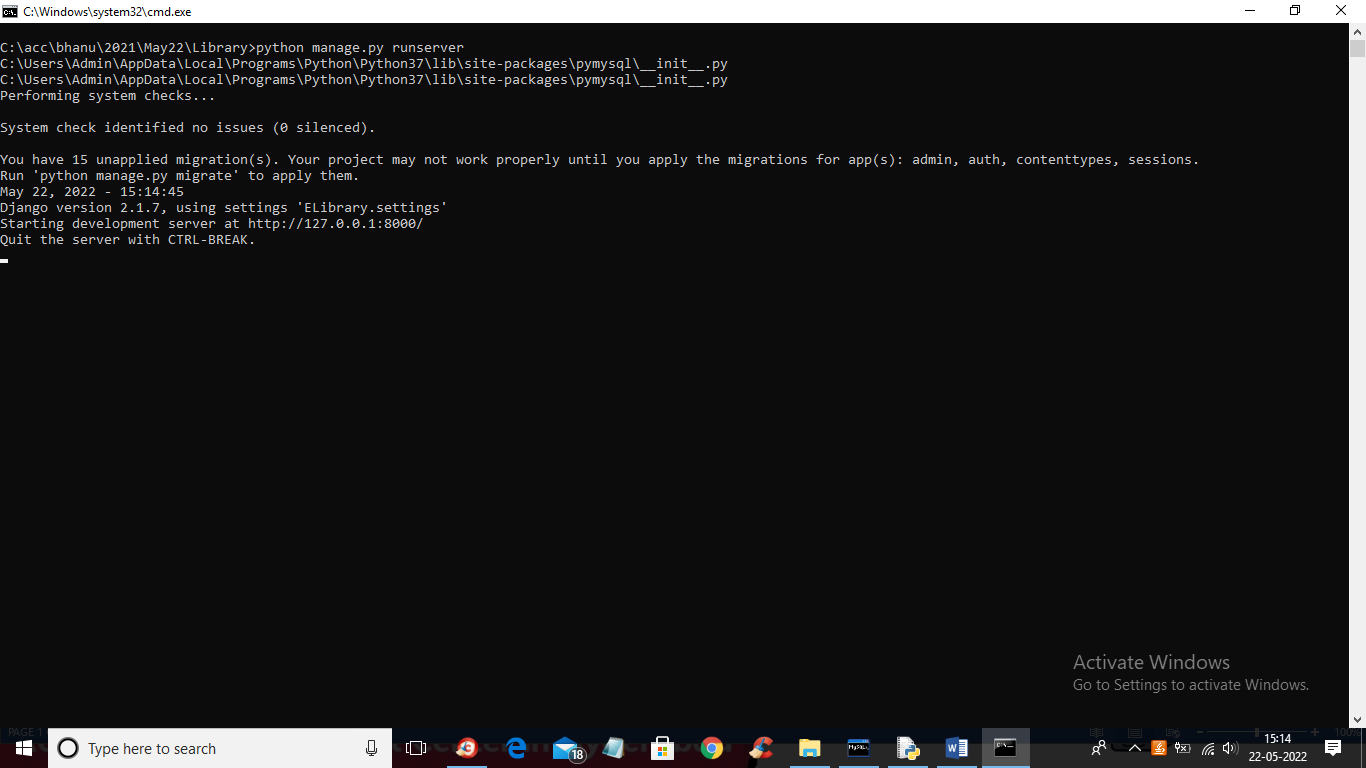
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

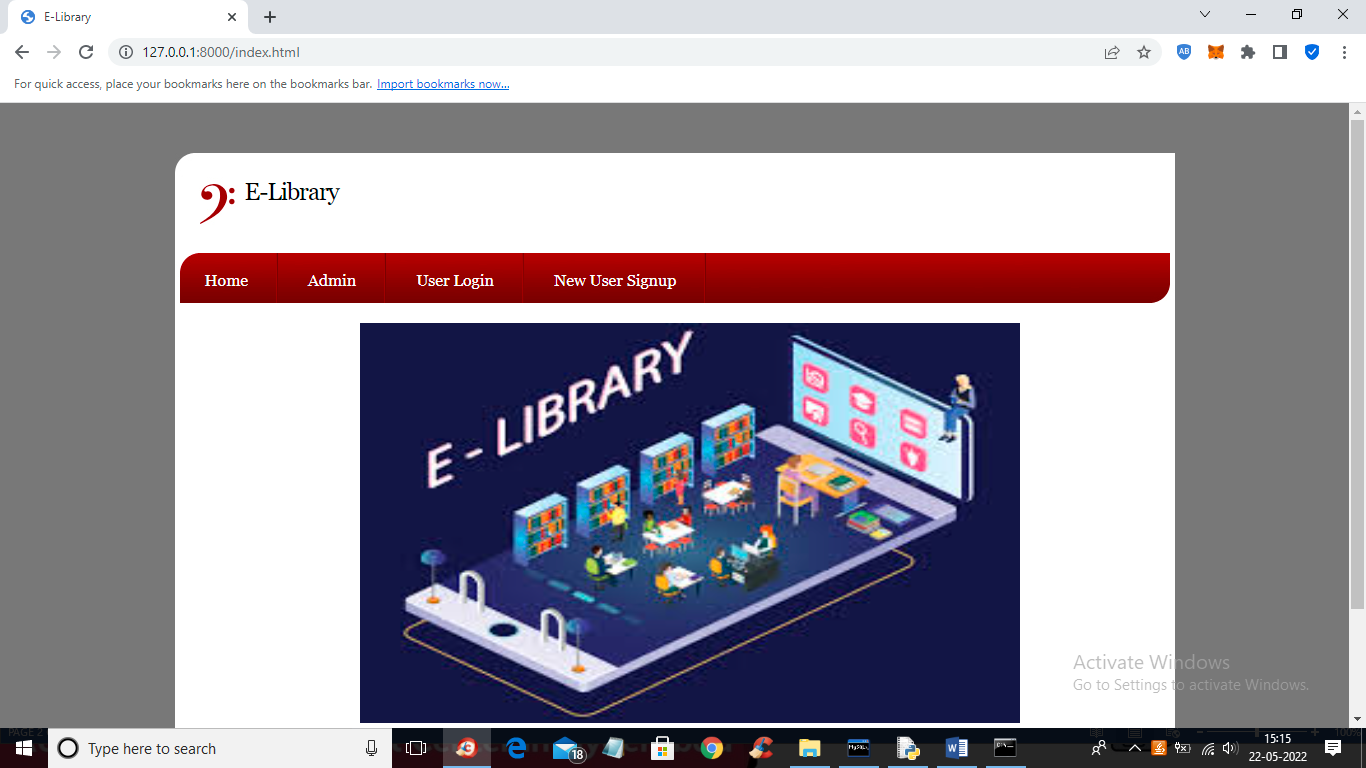
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | | **Actual** |
| 01 | Admin login | Verify  Admin login or not | If Admin may not login | we cannot do any further operations | we can do further operations | | High | High |
| 02 | Student login | Verify  Student login or not | If Student may not login | we cannot do any further operations | we can do further operations | | High | High |

**7. SCREENSHOTS:**

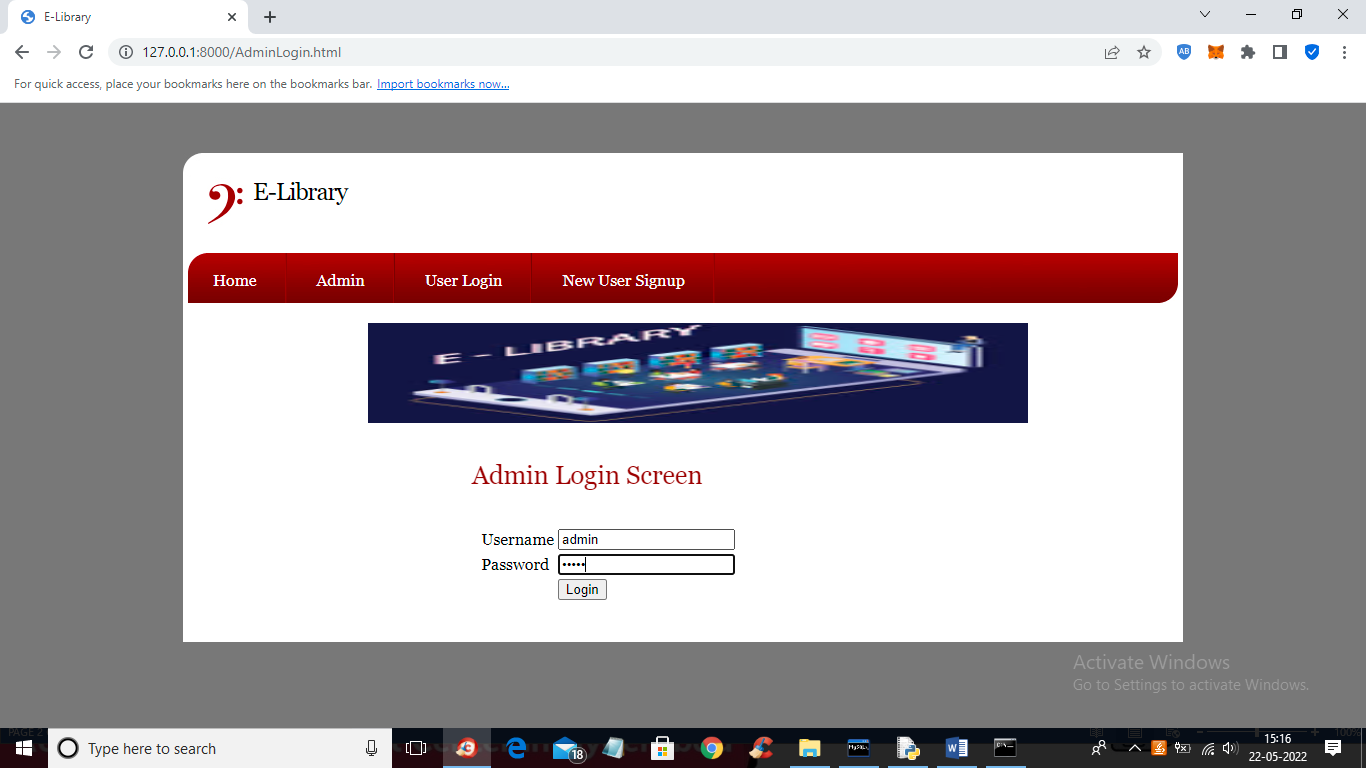
To implement this project, we have designed following modules



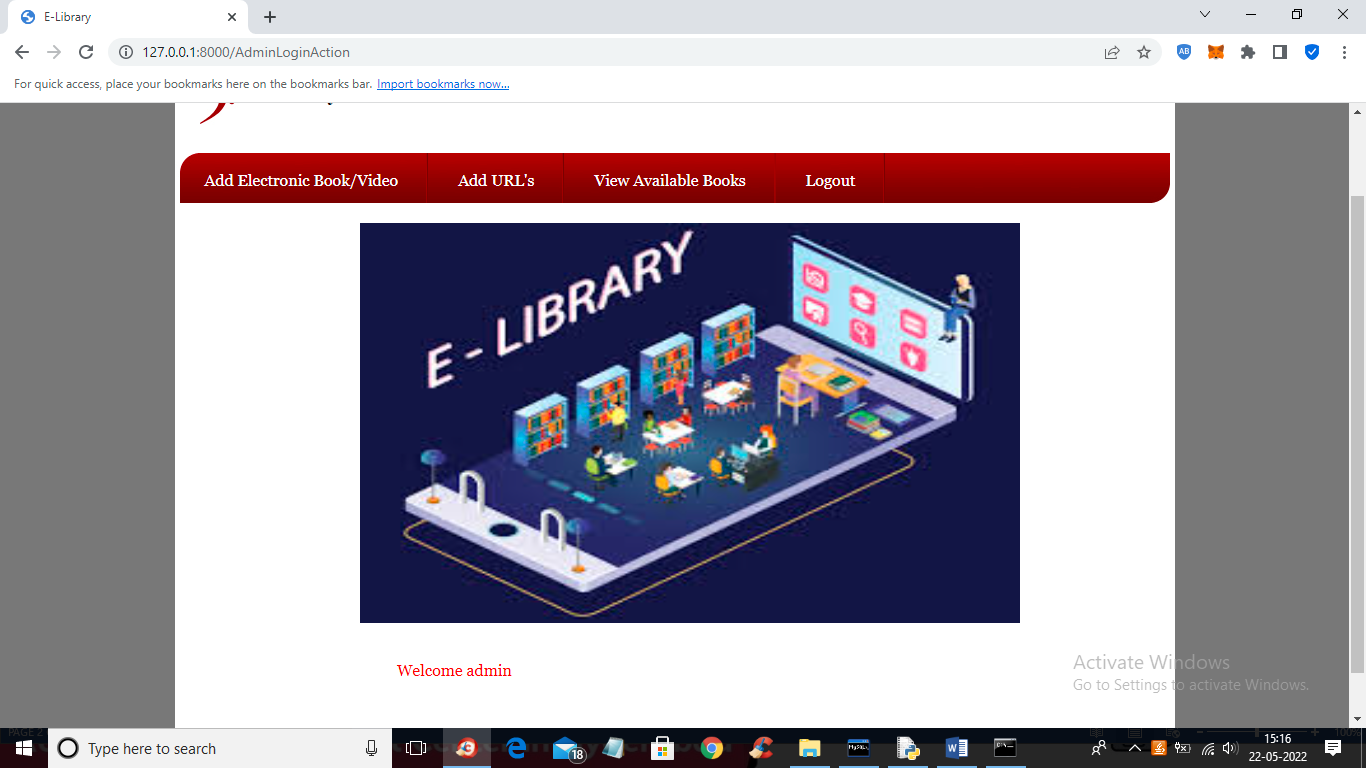
In above screen python DJANGO server started and now open browser and enter URL as http;//127.0.0.1: 8000/index.html and press enter key to get below screen



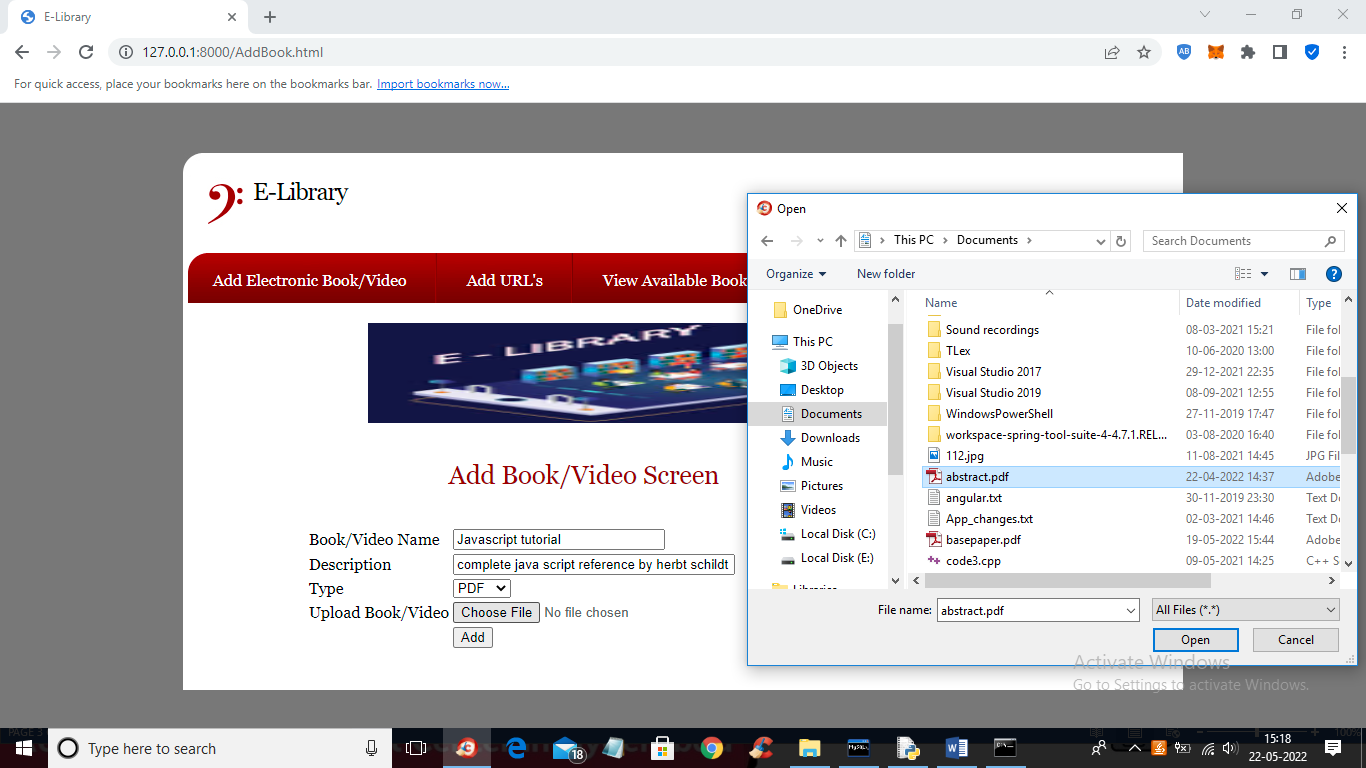
In above screen click on ‘Admin’ link to get below login screen



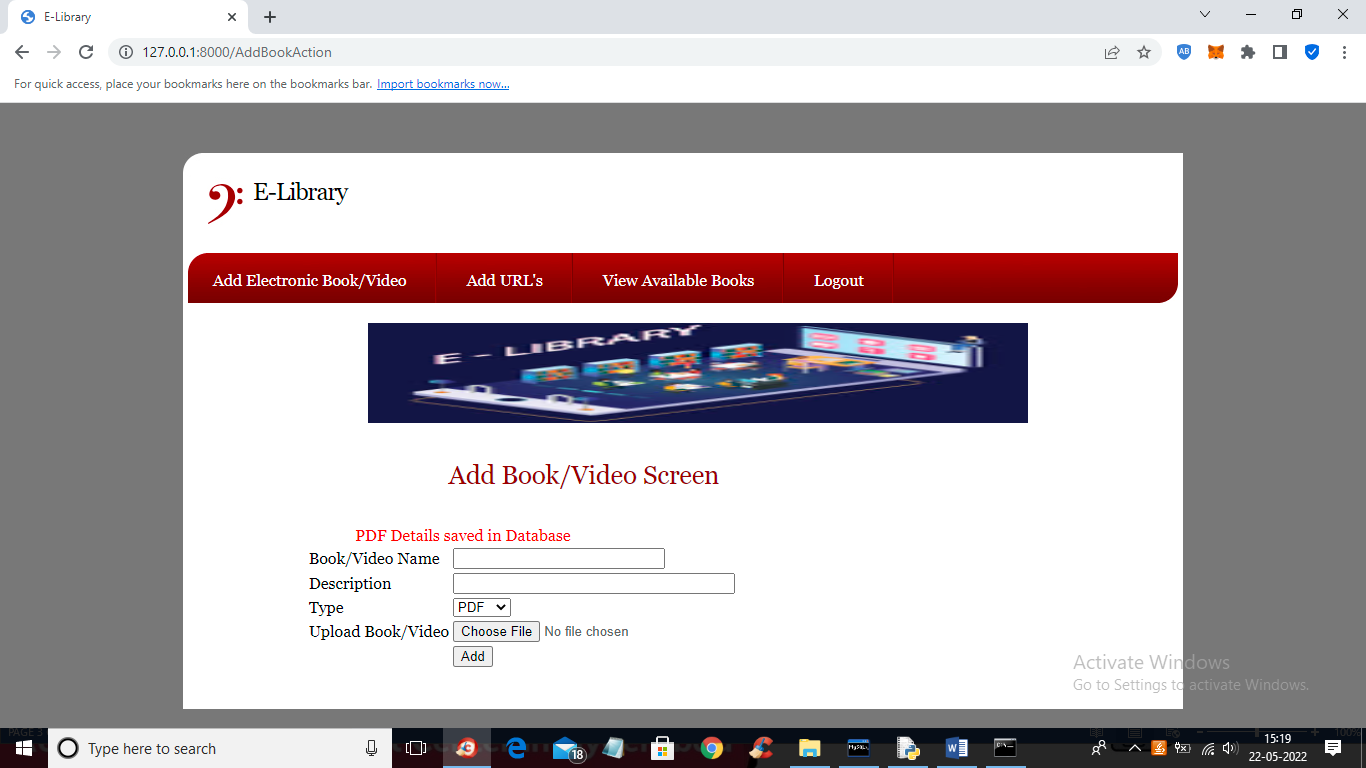
In above screen admin is login and then click on button to get below screen



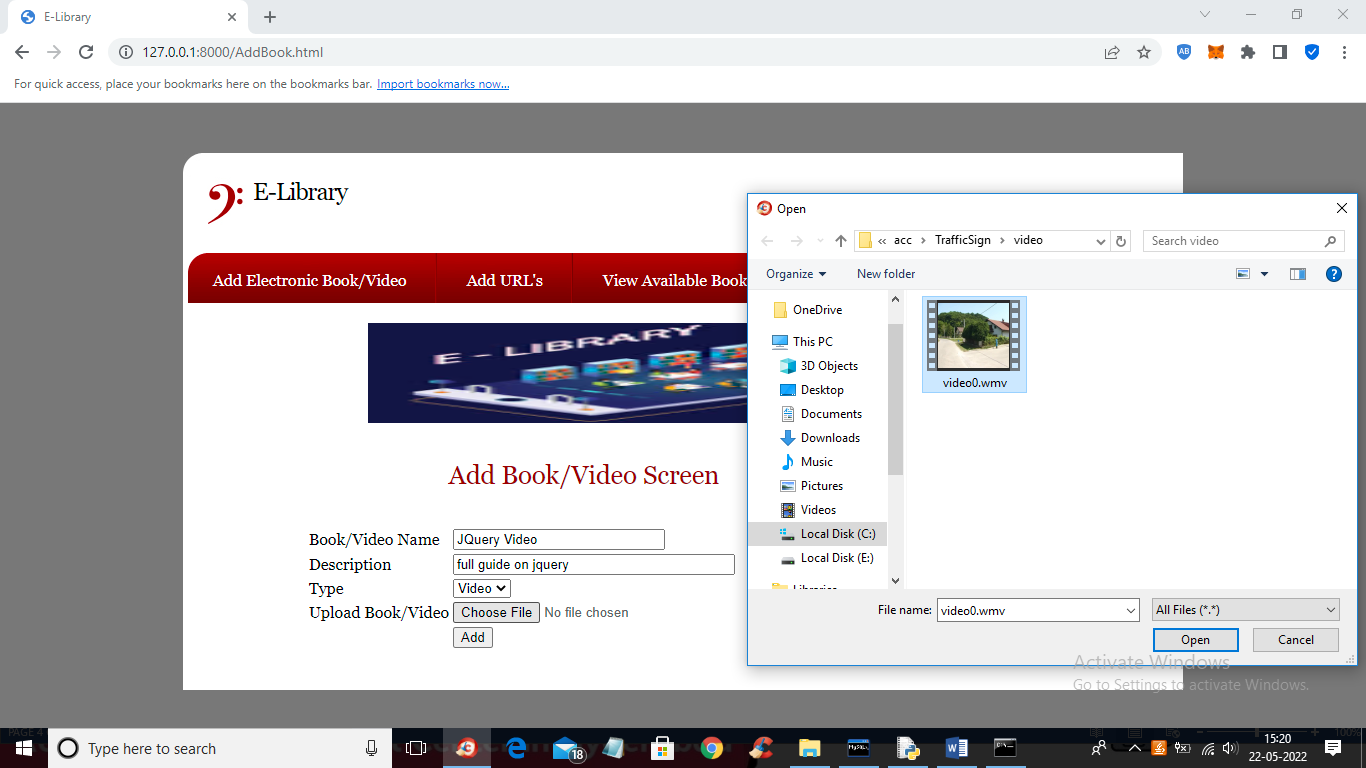
In above screen admin can click on ‘Add Electronics Book/Video’ link to add book or video



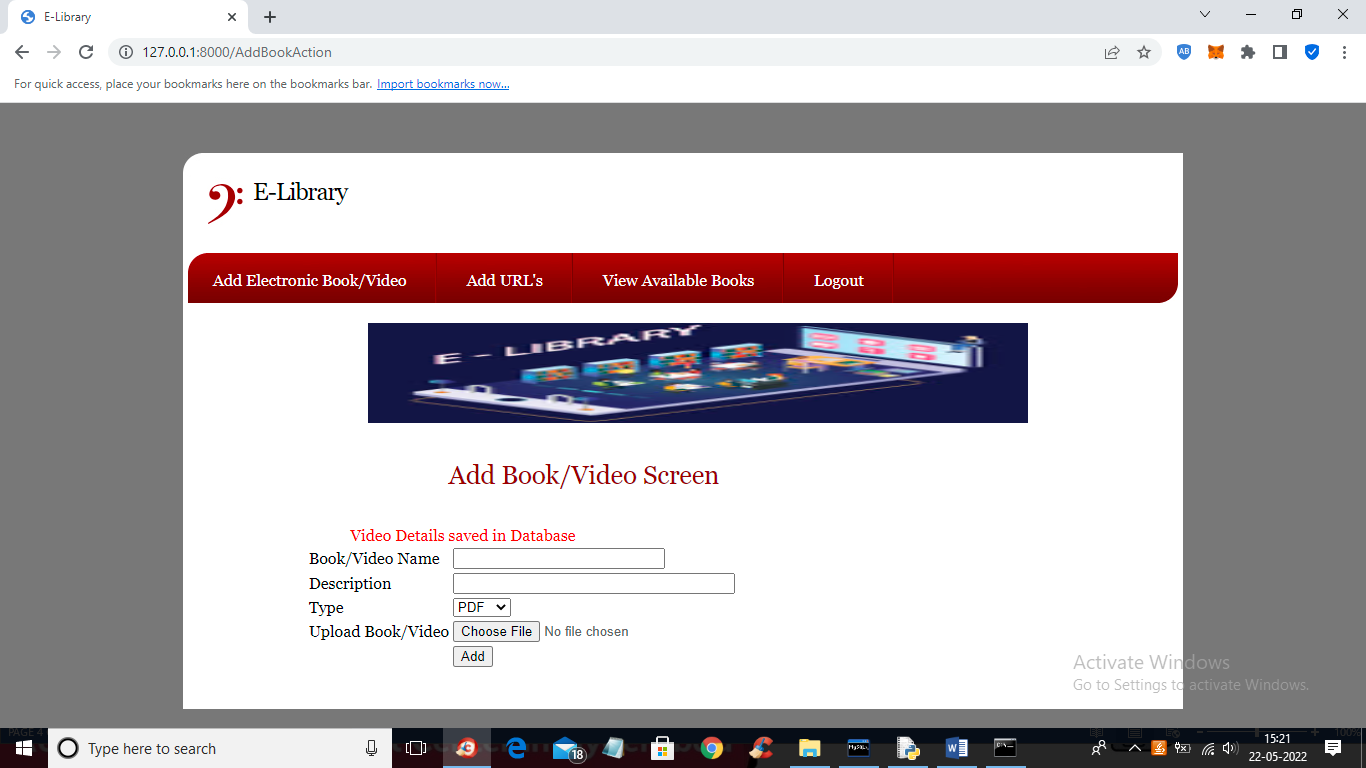
In above screen admin is uploading PDF file and then click on ‘Open’ and ‘Add’ button to add book to database and similarly admin can upload video also



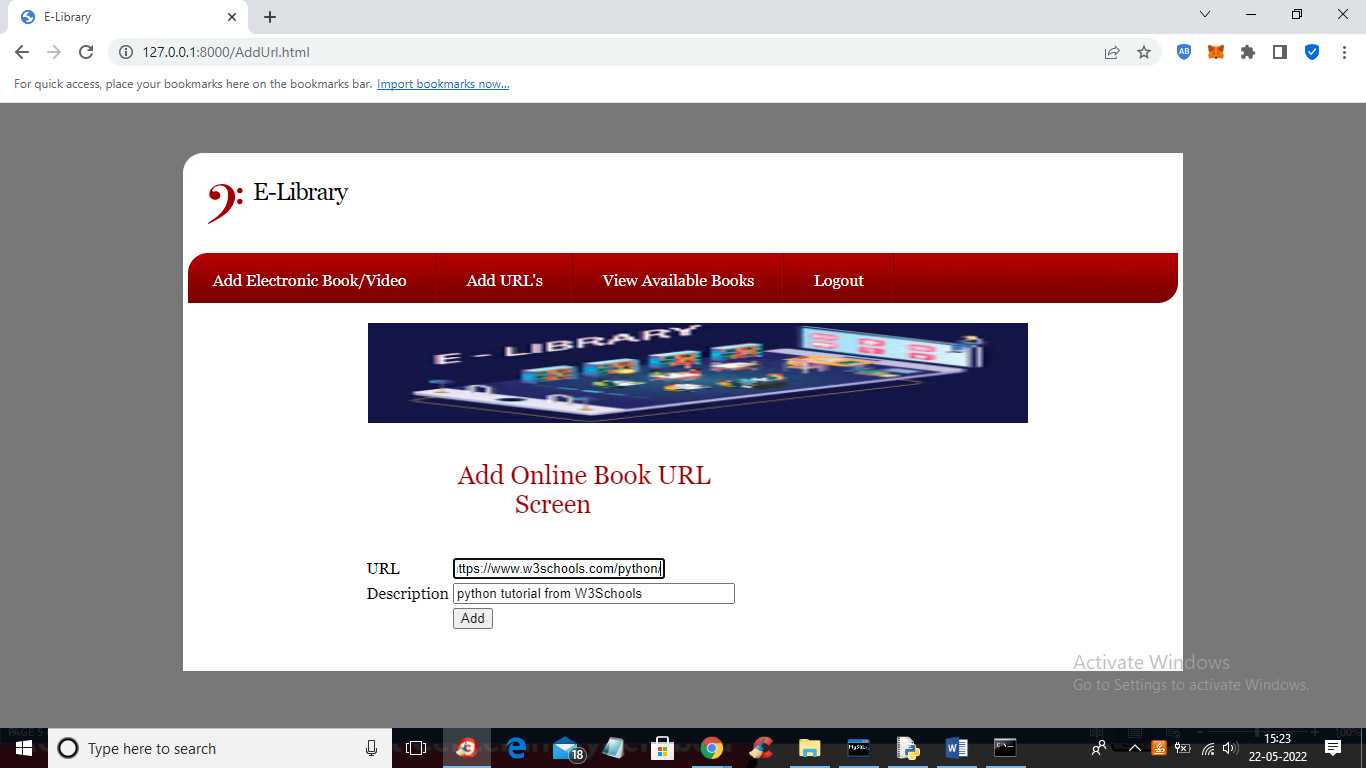
In above screen PDF details added to database and similarly you can add other books also



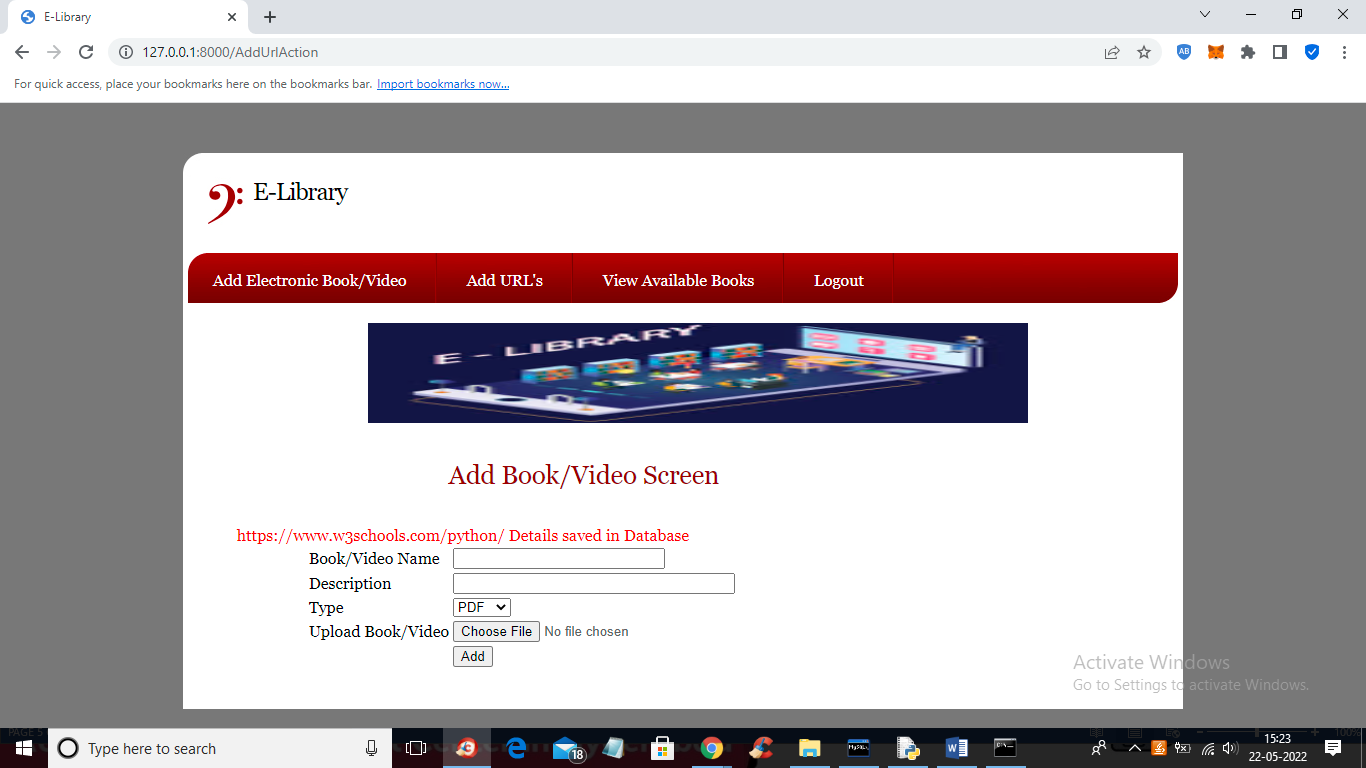
In above screen admin is uploading video tutorial and click button to add details



In above screen video details saved in database and now click on ‘Add URL’s’ link to add URL for tutorials



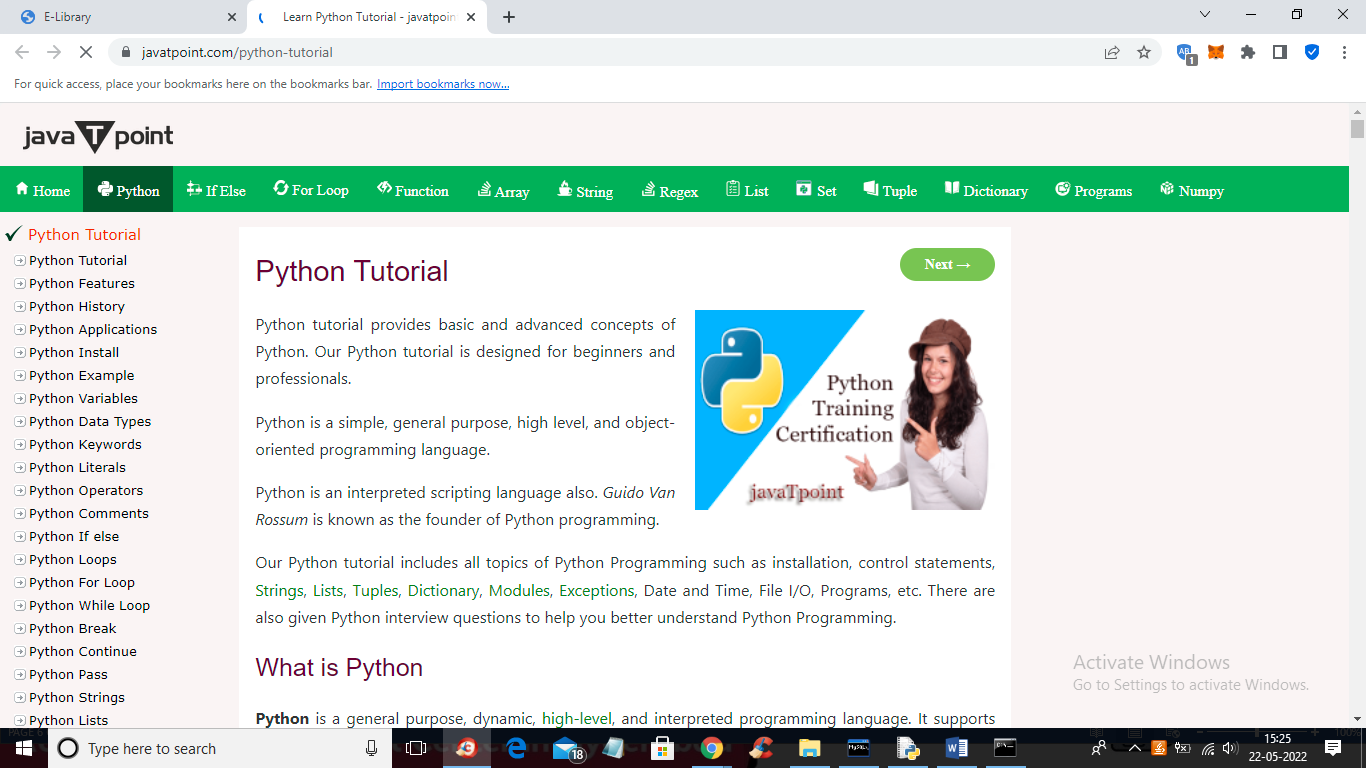
In above screen admin is adding URL tutorial and click button to get below output



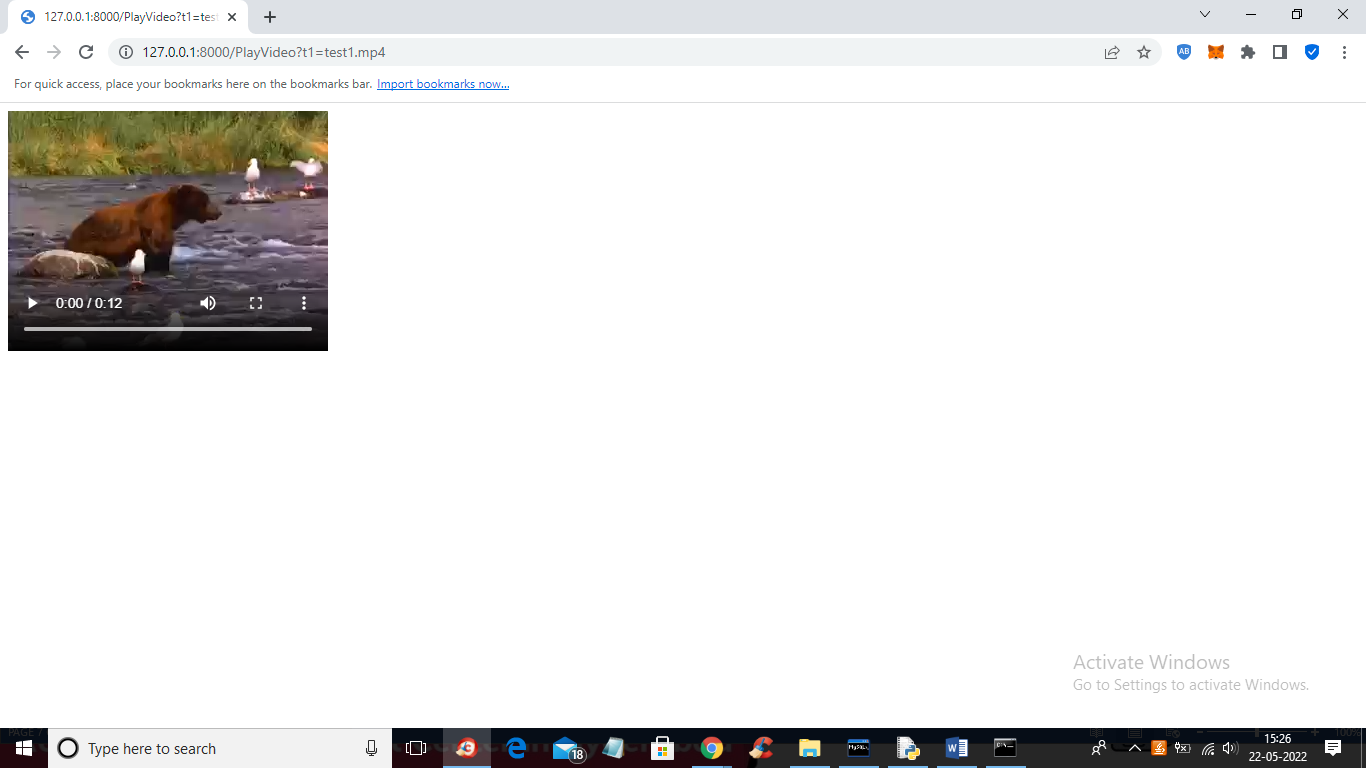
In above screen tutorial is added and now click on ‘View Available Books’ link to view below output



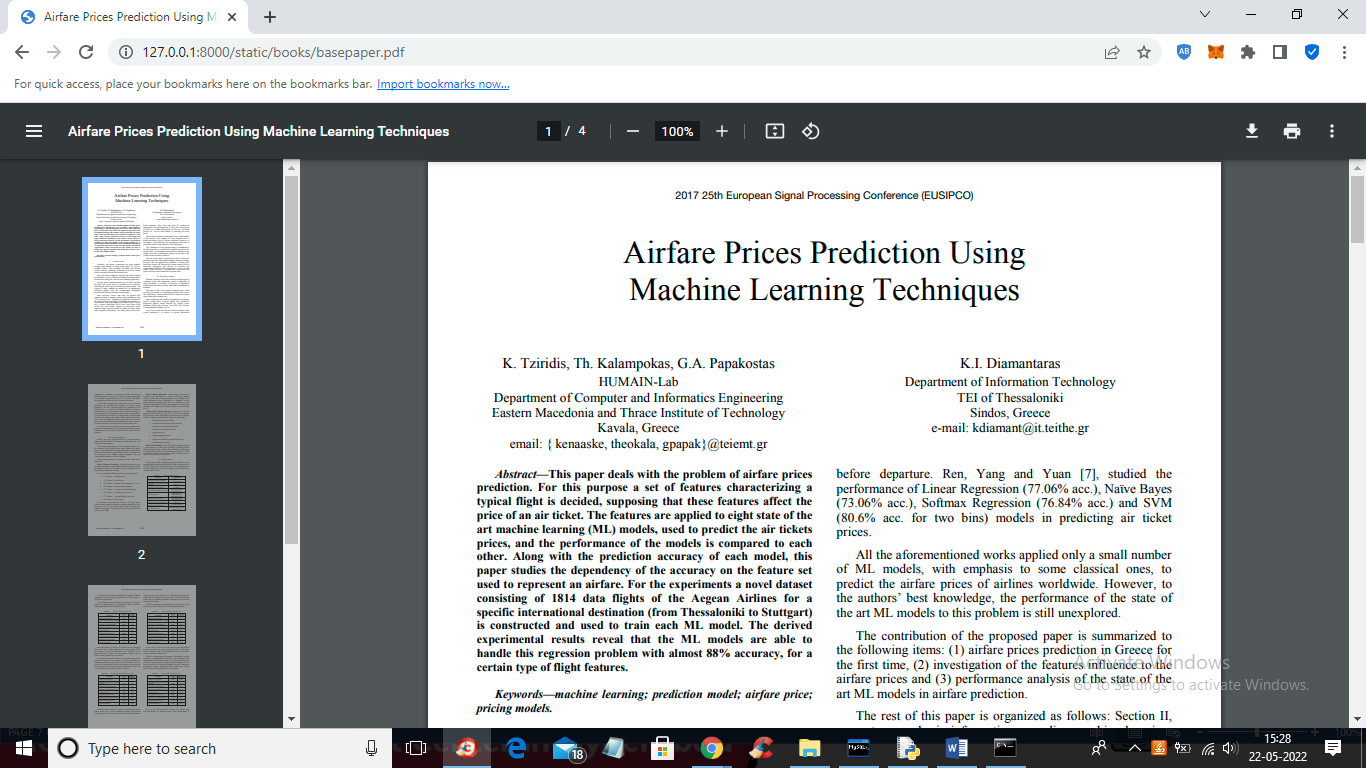
In above screen admin can view all book details and then click on desired image to view its tutorial or Click Here to delete that tutorial. So admin can add and delete tutorial



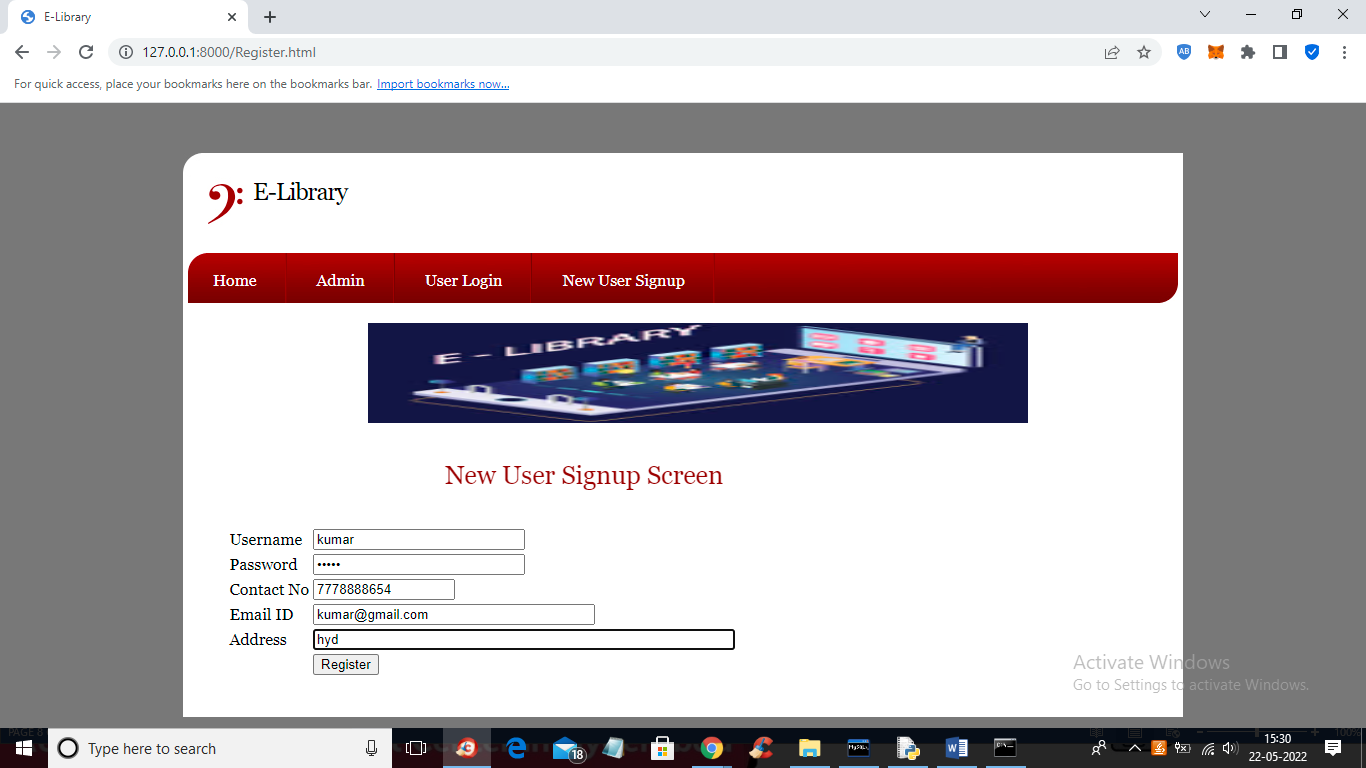
In above screen tutorial URL is open and in below screen we can see VIDEO



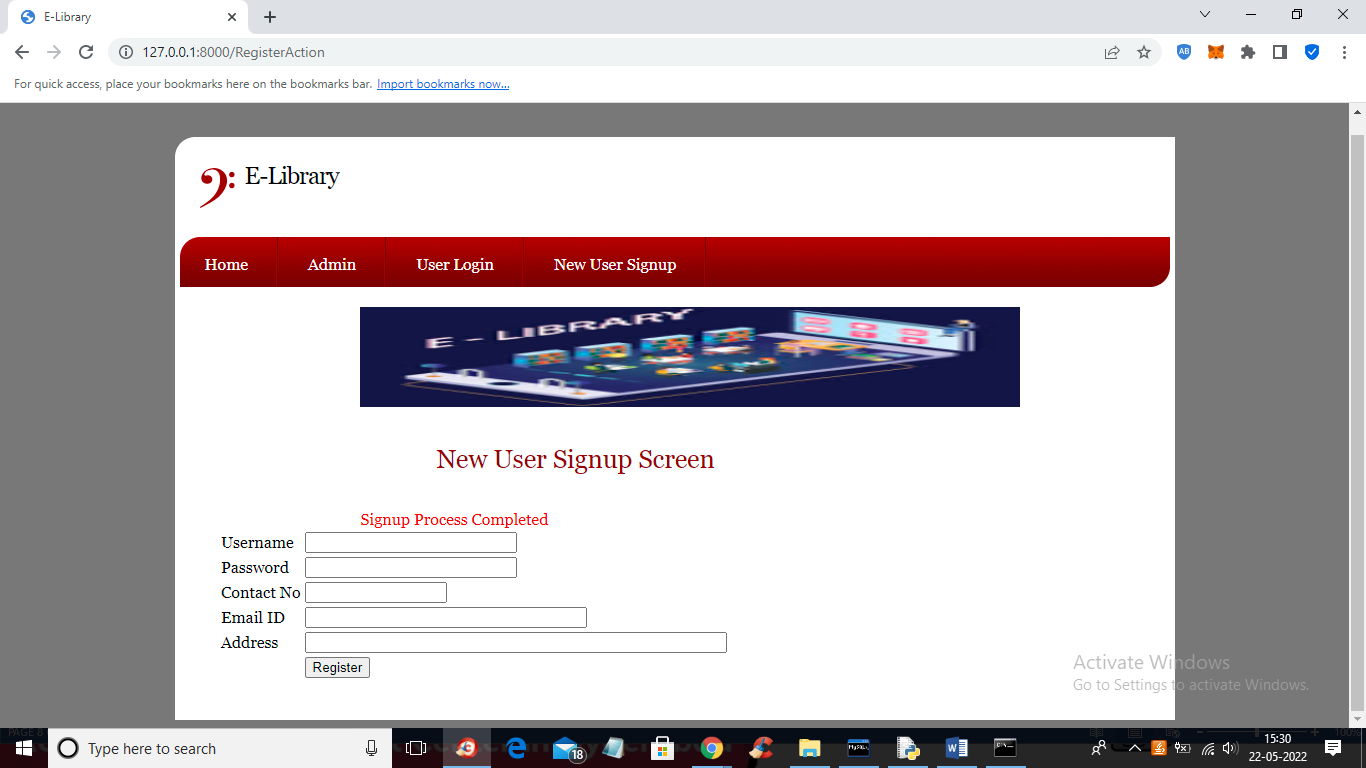
In below screen we can open PDF tutorial



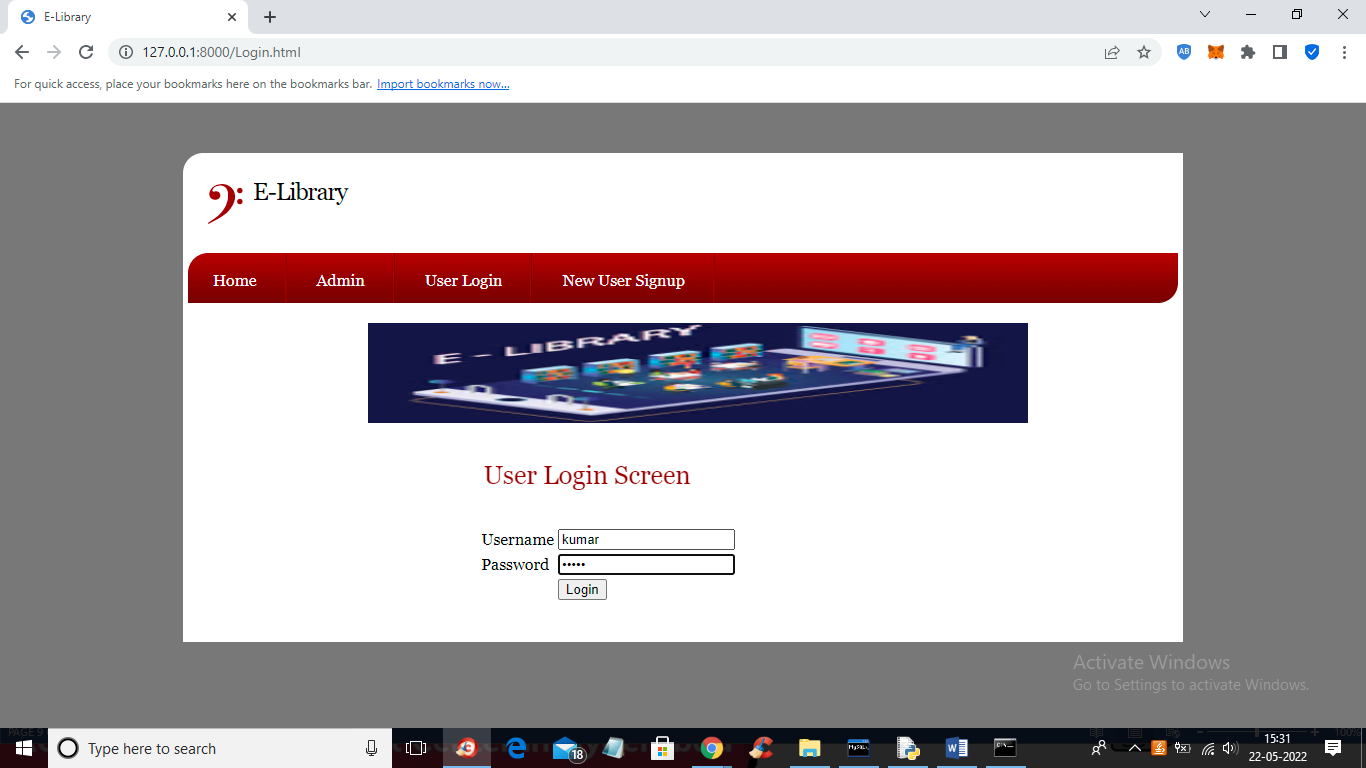
In above screen PDF tutorial is open and similarly admin can delete any tutorial and now logout and signup student



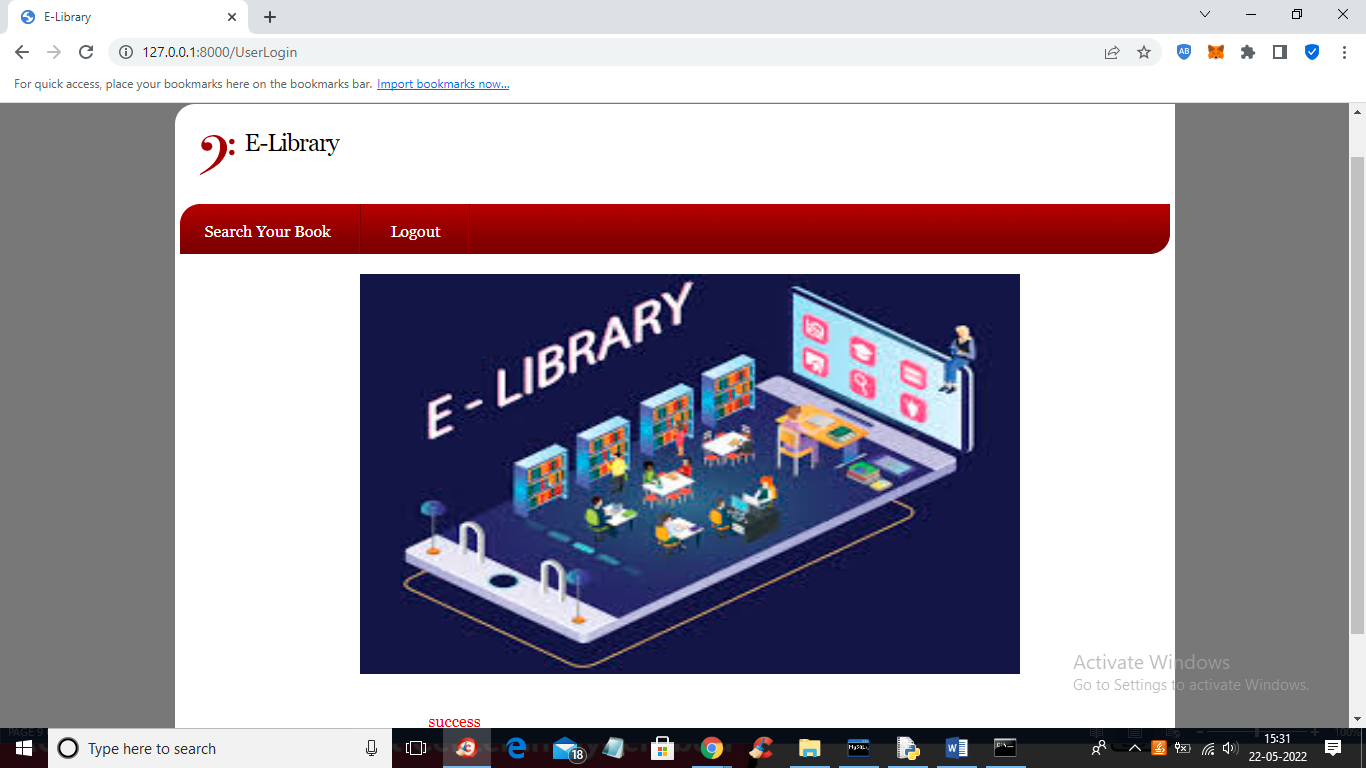
In above screen student is signing up and then click button to get below output



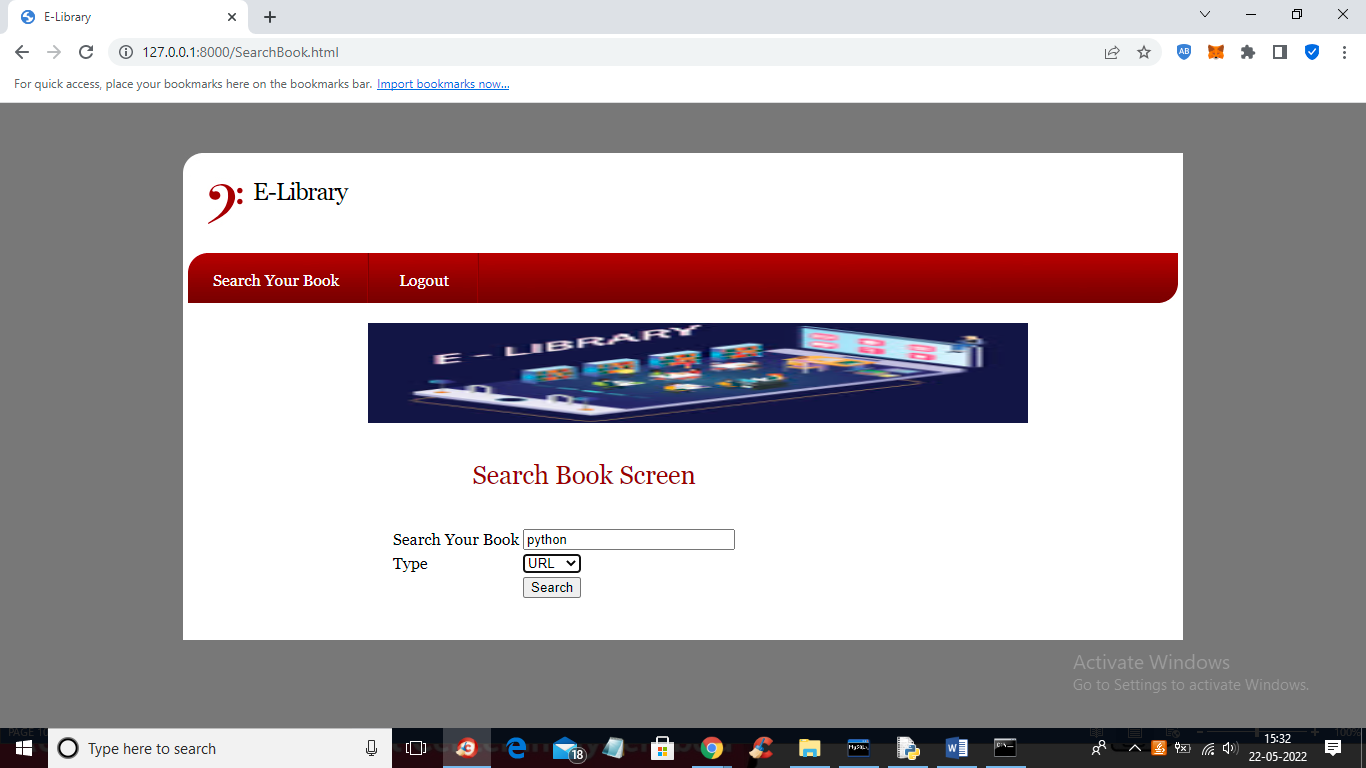
In above screen signup process completed and now click on ‘User Login’ link to get below screen



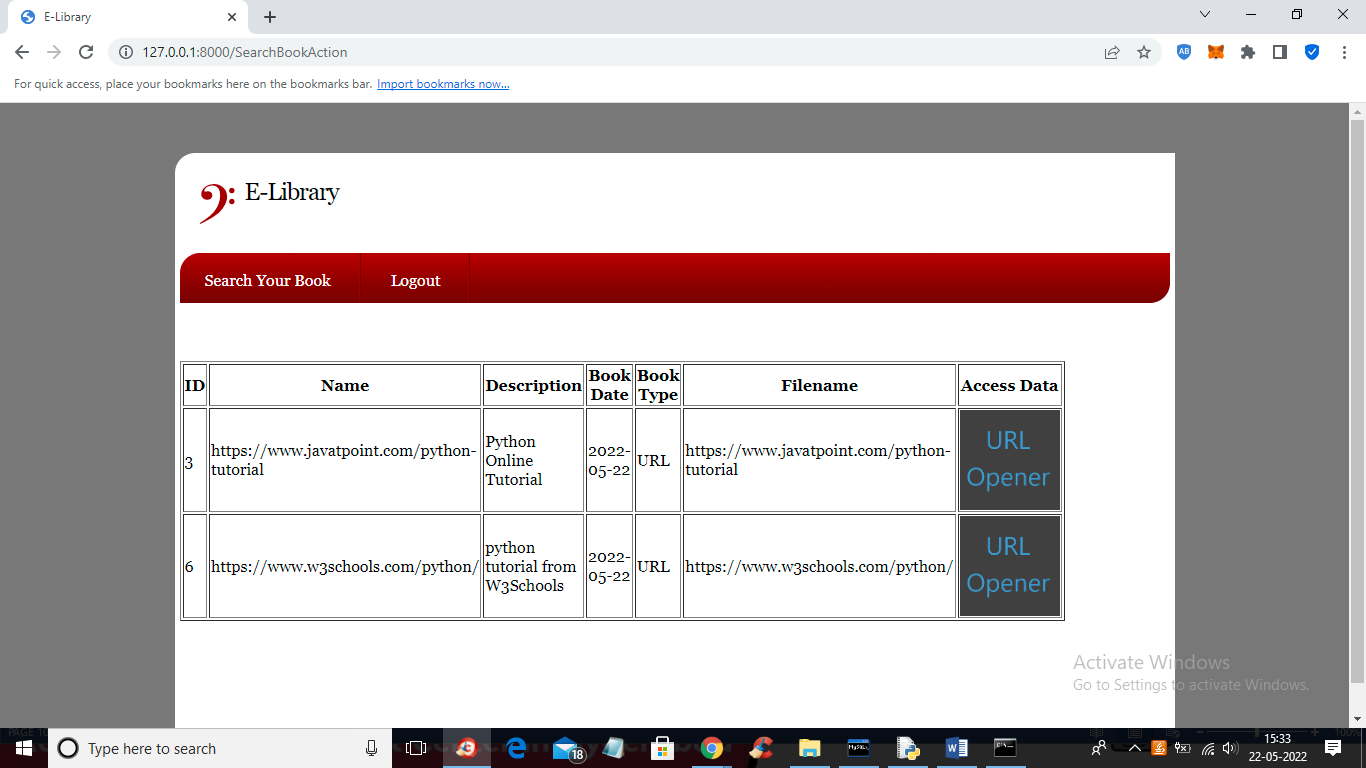
In above screen user is login and after login will get below screen



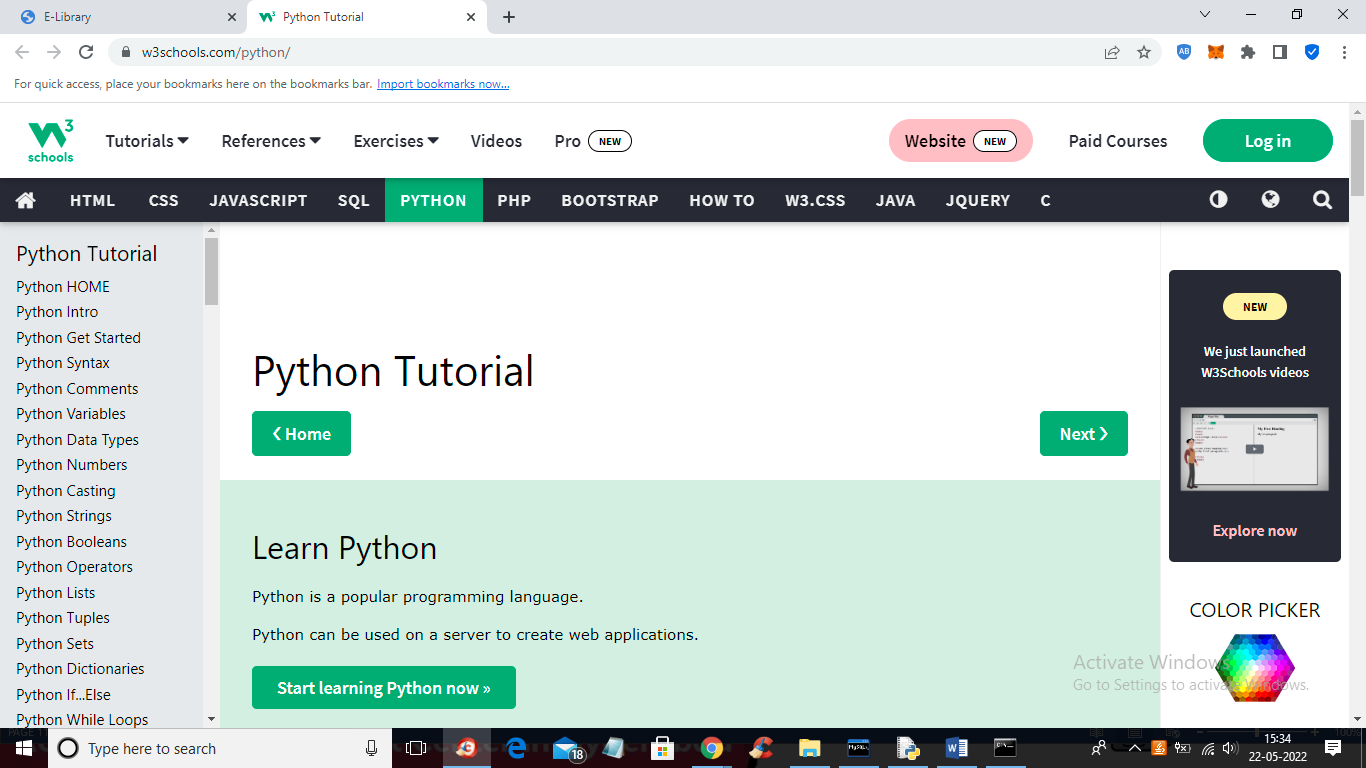
In above screen click on ‘Search Your Book’ link to get below screen



In above screen user is searching python tutorial with URL and get below output



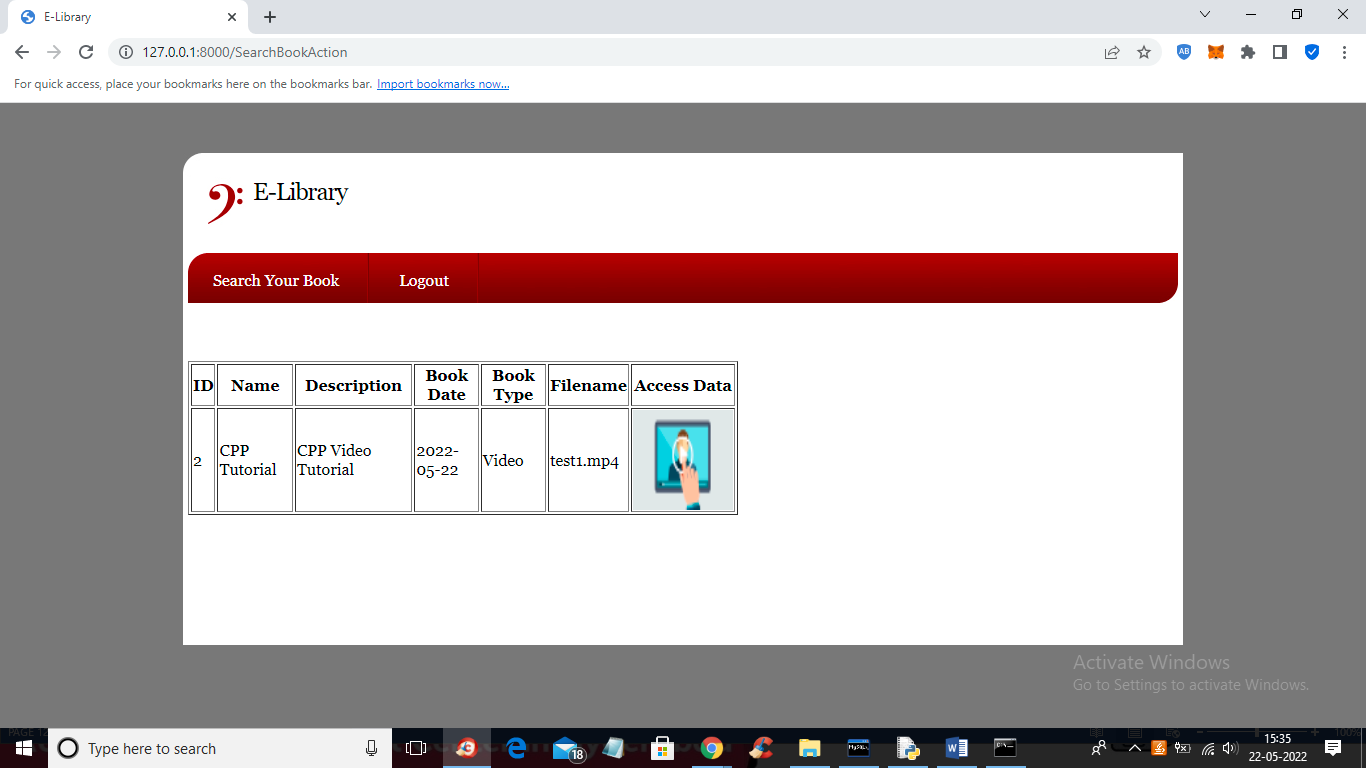
In above screen user will get all URLS for python tutorial and can click on ‘URL Opener’ link to visit that URL



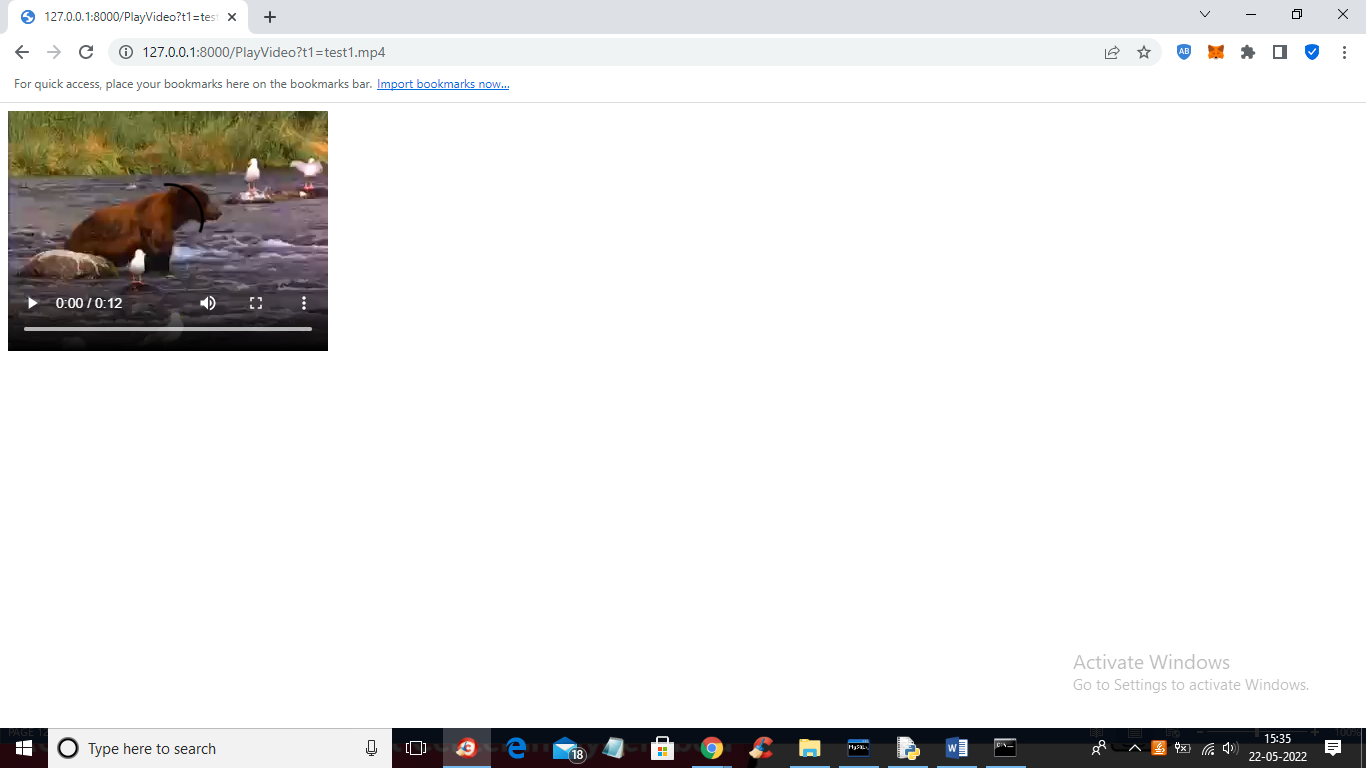
In above screen user can open and read tutorial from URL and in below screen I am searching another query



In above screen I am searching CPP video based tutorial and click button to get below output



In above screen user can click on ‘PLAY ICON” to play tutorial



In above screen tutorial is playing and similarly you can add and search any tutorial

**8. CONCLUSION:**

In most developing countries, most people have no access to quality sources of higher education. Additionally, the quality of the education that the population receives is typically suspect. For the present project of electronic library, the targeted users are located in Sub-Saharan African countries, a region that has felt the effects of “brain drain”, where the most talented go abroad to study and work and, rarely return to their homeland. Therefore, the quality of teaching and education may not meet the international standards. Several major constraints must be overcome to create good conditions for the development of science and technology in this undeveloped area. One major constraint is the lack of accessibility to electronic library. Electronic interface represents the only opportunity to provide library services and resources to online users anytime and anywhere. The web interface must be clear and uncluttered, easy to maneuver, and must provide built-in redundancy to accommodate different learning styles. Students and other college users must acquire the ability to locate, evaluate, and use information successfully. Electronic library projects do not require heavy investment. With applying an efficient project management technique, electronic library can be designed for the benefit of poor people who have not access to modern information systems and technology. Project management can also make the electronic library financially gainful with possible income from advertisement and other relevant services.

**9. REFERENCES:**

1] J. Ma, “Managing Metadata for Digital Projects”, Library Collections, Acquisitions, & Technical Services, vol. 30, pp.3-17, 2006.

[2] H.-L. Chen and G. Choi, “Construction of a Digital Video Library: A Socio-Technical Pilot Study on College Students’ Attitudes” The Journal of Academic Librarianship, Vol. 31, N5, pp. 469-476, 19 July 2005.

[3] C. Gutwin, G. Paynter, I. witten, C. Nevill-Manning, E. Frank, “Improving Browsing in Digital Libraries with Key phrase Indexes”, Decision Support Systems, vol. 27, pp. 81-104, 1999.

[4] M. S. Woodley, “A Digital Library Project on a Shoestring”, Library Collections, Acquisations, & Technical Services 26(2002) 199-206.

[5] E. J. Shaw, “Building A Digital Library: A Technology Manager’s Point of View”, The Journal of Academic Librarianship, Vol. 26, Number 6, pp. 394-398.

[6] K. Calhoun, “From Information Gateway to Digital Library Management System: a case Analysis”, Library Collections, Acquisitions, & Technical Services, vol. 26, pp. 141-150, 2002.

[7] R. J. Heckant, “Imagining the Digital Library in a Commercialized Internet”, the Journal of Academic Librarianship, Vol. 25, Number 4, pp. 274-280; July 1999.

[8] L.-S. Chen, “Design and Implementation of Intelligent Library System”, Library Collections, Acquisitions, & Technical Services, vol.. 32, pp. 127-141, (2009.

[9] C. Schwartz, “Digital Libraries: AN Overview”, The Journal of Academic Librarianship, Vol. 26, N. 6, pp. 385-393, November 2000.

[10] L. Feng, M. A. Jeusfeld, J. Hoppenbrouwers, “Beyond Information Searching and Browsing: Acquiring Knowledge from Digital Libraries”, Information Processing and Management, vol. 41, pp. 97- 120, 2005. [11] H. I. Xie, “Users’ Evaluation of Digital Libraries (DLs): their Users, their Criteria, and their assessment”, Information Processing and Management, vol. 44, pp. 1346-1373, 2008.

[12] A. Shtub, J. F.Bard, S. Globerson, Project Management: Processes, Methodologies, Economics 2nd Ed., Pearson Prentice Hall, 2006.

[13] J. A. Lawrance Jr. and B. A. Pasternack, Applied Management Science 2nd Ed, John Wiley & Sons, 2002.