Project Report on

Digital Key of Digital India

A Dissertation

Submitted by:

Mansi Yadav	**210550181064**
Kadam Rushikesh Ganesh	**210550181048**
K N Rakshith Raju	**210550181047**
Shiv Mangal Yadav	**210550181101**
Satyabrata Barik	**210550181097**

Under the guidance of

Ms. Vathsala H

(CDAC Designation)

Submitted in partial fulfilment of the course of

e-Diploma in Advance Computing

from

CDAC ACTS

Knowledge Park, Bengaluru

CERTIFICATE

This is to certify that

Mansi Yadav	**210550181064**
Kadam Rushikesh Ganesh	**210550181048**
K N Rakshith Raju	**210550181047**
Shiv Mangal Yadav	**210550181101**
Satyabrata Barik	**210550181097**

Has successfully completed the project on

Digital Key of Digital India

Satisfactorily in partial fulfilment of the requirement of **e-Diploma in Advanced Computing** at Centre for Development of Advanced Computing(CDAC), Bengaluru.

Ms. Vathsala H
(Project Guide)

Ms. Uma Prasad
(Course Coordinator)

ACKNOWLEDGEMENT

Completing a task is never a one-man effort. It is often the result of valuable contribution of a number of individuals in a direct or indirect manner that helps in shaping and achieving an objective. It is very difficult for anyone to complete a project without the active cooperation and the benefit of the advice from the people who are experts in their field of specialization. The satisfaction and euphoria that accompanies the successful completion of any task would not be complete without the mention of the people who made it possible with due honour, we want to thank all the personalities who made us able to do this interesting work. We have taken this opportunity to express our deepest sense of regard and gratefulness to **Ms. Vathsala H** whose valuable guidance, heartening encouragement and sympathetic attitude every step has steered this project to its successful completion.

We express our profound thanks to our faculty members of CDAC Bangalore who cultivated devotion and determination in us which have been a deep source of wisdom and inspiration for such initiative.

We are sincerely grateful to **Ms. Vathsala** for her assistance and guidance through the study.

DATE: 29th Sept 2021

Name	PNR No.
Mansi Yadav	**210550181064**
Kadam Rushikesh Ganesh	**210550181048**
K N Rakshith Raju	**210550181047**
Shiv Mangal Yadav	**210550181101**
Satyabrata Barik	**210550181097**

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER -1

INTRODUCTION

In present time everyone has either heard or talking about the thing i.e., "**DATA IS THE NEW OIL**", so in this new era of technological advancement collecting data and processing it is the need of the hour. And with the exponential growth of the global technologies our India is also growing on a par to the rest of the world. And to support this the Government of India has also launched a campaign, **Digital India**. Our project is a value addition towards digitalizing the documents in such a way that it will serve as a refence to other documents.

1.1 PURPOSE

In this era of internet everyone is using the digital network world for entertainment, socializing, to research, to study, to develop some software and a lot many things. Our project i.e., **Digital Key of Digital India** will help those who want to study or do research by providing them references of their field of interest. This project will help everyone in widening their knowledge by helping them with references or descriptions. As the development of any nation depends on the minds of the nation so it will help India to grow faster. The very purpose of this project is to provide the user references for their documents so that they can easily understand the contents and also to digitalized document in such a way that it can be shown in browser as HTML pages.

1.2 PROBLEM DEFINITION

This project is for providing user a reference to their documents in such a way that words which are out of scope or difficult to understand in their document will be hyperlinked to a book related to the same topic that of the user's documents. By this he can see the proper explanation of the difficult words he/she encounter and can easily understand the context.

1.3 EXISTING SYSTEM

The existing system for finding references is to be done manually by the user. She/He may refer to some hardcopy books, journals or she/he can search it over internet which is a time-consuming task and also depends on other factors like internet speed, availability of reference material, etc.

1.3.1 The main problems with the existing system: -

- Manual or semi manual system itself describe as an outdated.
- There is no one stop solution for finding references from a centralized system.
- It is very difficult to store physical books at one place.
- In the present era of internet advancement everyone is moving towards digital data rather than available data in the form of books, journals, etc.
- Searching correct information manually from the ocean of data is a very difficult task.
- Referring to hardcopy can damage the document itself due to multiple usage and also the readability of document might not be proper due to multiple times usage.
- More time consuming.
- Not reliable and effective.
- There is always a chance of human error which is to minimized as much as possible.

1.4 PROPOSED SYSTEM

Digital Key of Digital India provide the user a way to search on their desired topic automatically just by uploading their content as pdfs. Here user will have to upload her/his reference book and the book's content page separately and after that she/he has to upload the desired document which she/he wants to study or whose references she/he want to generate.

By doing this user will get his content in the browser and the words which will match with the headings of the uploaded reference book, will be hyperlinked to the headings of the book.

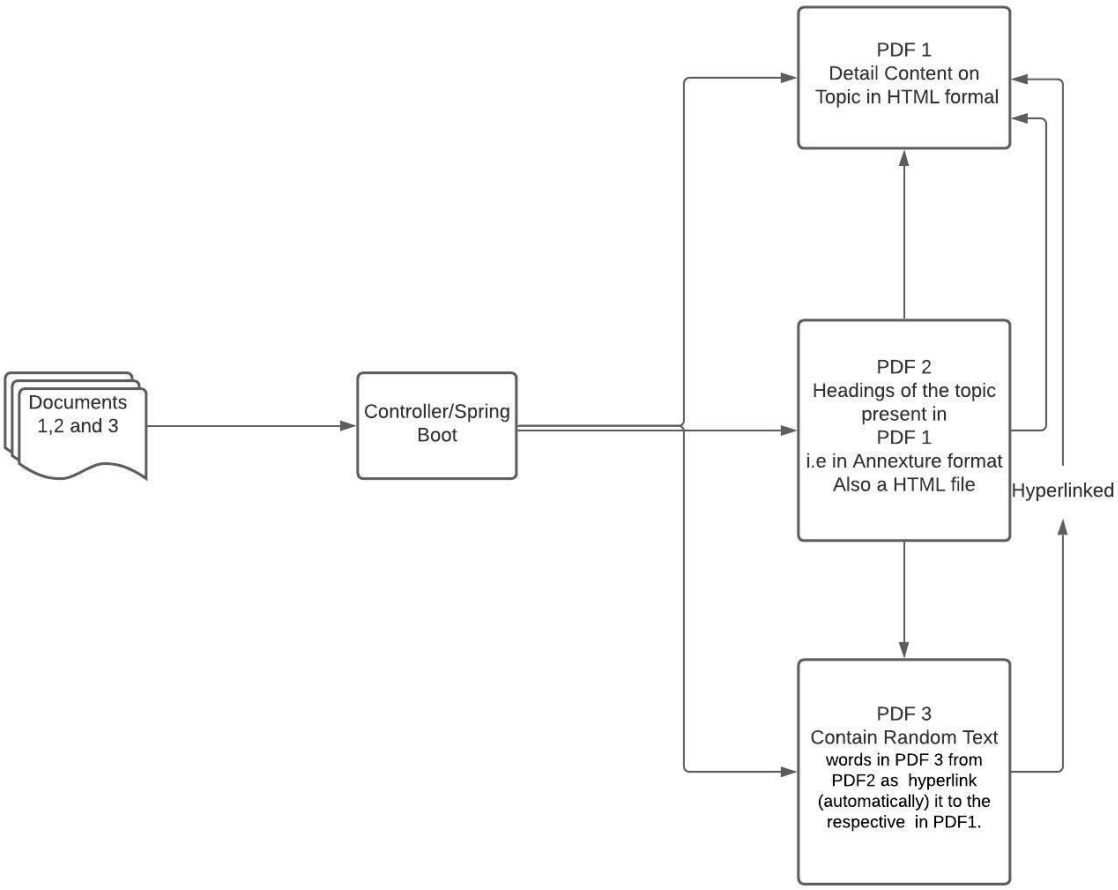


Fig. No. 1.2 (Proposed System DFD)

1.5 SCOPE

Developing a web app which can create references dynamically and automatically has a huge amount of scope in this exponentially expanding digital world. It will digitalized and automate huge amount of books which can be very easy accessed by the user. It will help the user to find her/his required references by filtering the unwanted data.

Furthermore, in the field of research and development it can help the brightest minds of our country to grow together by sharing only the intended information.

By the help this project anyone can access his desired data automatically in real-time.

It will help the students, teachers, professional, researchers, etc to access their required information from the vast ocean of information in a very simple way which will leads to the growth and also intimately growth of our country, INDIA.

1.6 SDLC Used

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. And in this software development project we have used waterfall model.

1.6.1 Waterfall Model

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

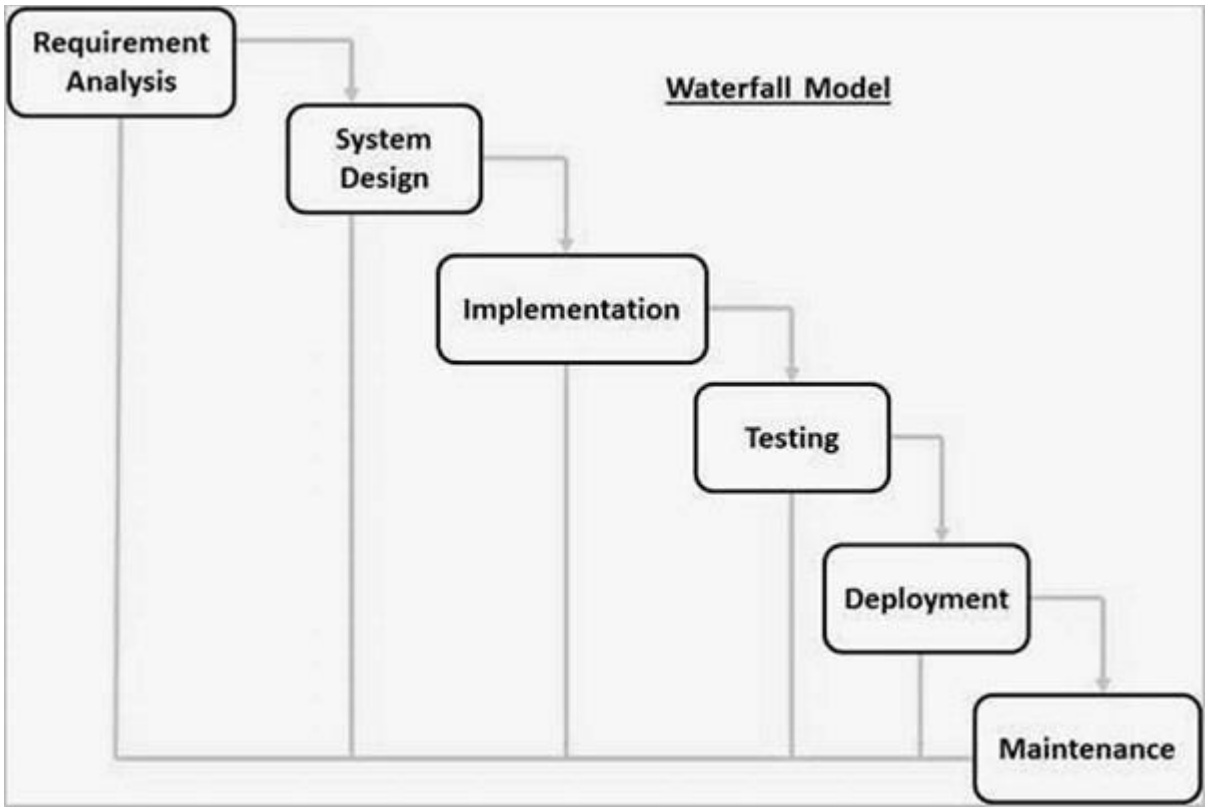


Fig. No. 1.2

The sequential phases in Waterfall model are −

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

CHAPTER -2
FEASIBILITY STUDY

FEASIBILITY STUDY

The primary objective of this feasibility report is to inform the objective of proposed system would be to overcome the problem faced in the manual or semi-digital system. Searching for references whether online or offline would not be a problem and also filtering of required information would not be required in the proposed system. There is an increased risk of error in the manual system. Redundancy of data creeps in the manual system and it becomes a very time-consuming job to search information criteria related to the required topic. This burden can be reduced by digitalizing and automating the whole process.
The user about following matters:

- What are the problems with conventional (manual) system?
- What are the goals and sub goals of the new system?
- What the proposed system would achieve?
- What will be the requirements for this achievement?
- Who will be involved in operating the system?
- The benefits, the system will give over conventional (Manual) System?
- The estimated cost of implementation?

2.1 Technical Feasibility:

It involves determining whether or not a system can actually be constructed to solve the problem at hand. Some user expects too much of computer, assuming that computers can accurately predicted the future, immediately reflect all information in an organization, easily understand speech, or figure out how to handle difficult problems. The technical purchases raised during the feasibility stages of the investigation are:

- Does the necessary technology exist (can it be acquired) to do what is suggested?

- Does the proposed equipment have the technical capacity to hold the data required to use the new system?
- Will the proposed system and components provide adequate responses to enquiries, regardless of the number or location of users?
- Can be system expanded, if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

For examples, if the proposal includes a printer that prints at rate of 4,000 lines per minute, a brief search shows that this is technically feasible. Whether it should be including in the configuration because of its cost is an economic decision. On the other hand, if user is requesting audio input to write, read, and change stored data, the proposal may not be technically feasible. I had found in my analysis that there are quite sufficient technical resources available at the organization and capable of handling user requirements.

2.2 Economical Feasibility:

It involved estimating benefits and costs. These benefits and cost may be tangible or intangible. Tangible benefits may include deceasing salary costs (by automating manual procedures), preventing costly but frequent errors, sending bill earlier in the month, and increasing control over inventory levels. Such benefits may directly estimate in rupees without much trouble. Tangible cost is easily estimated. Intangible benefits may include increasing quality of goods produced, upgrading or creating new customer, and developing a better understanding of the market. The economic and financial questions raised by analysts during the preliminary investigation seek estimates of:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.
- The benefits in the form of reduced costs or fewer costly errors.
- The cost if nothing changes (the system is not developed). Because of already availability of computer software and hardware in the organization it is economical feasible.

2.3 Time & Operational Feasibility:

Proposed projects are of course beneficial only if they can be turned into information systems that will meet the organization operating requirements. Simply stated, this test of feasibility asks if the system will work when developed and installed. Here are questions that will help test the operational feasibility of a project:

- Is there sufficient support for the project from the management and from users? If the current system is well liked and used to the extent that persons will not see reasons for a charge, there may be resistance.
- Have the users been involved in the planning and development of the project? Early involvement reduces the chances of resistance to the system and charge in general, and increases the likelihood of successful projects.
- No qualified and trained person is required for this system because all the transactions and entry are through integrated development environment so it is quite easy to operate.

2.4 Behavioural Feasibility:

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover, transfers, retraining, and changes in employee job status. Therefore, it is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business. It answers the following questions:

- **Is the audience likely to adopt the behaviour?** Is the current behaviour seen as a problem? How engrained or "rewarding" are the current or competing behaviours?
- **How costly is it** (time, effort, resources) for the audience segment to perform the behaviour?
- **How complex is the behaviour** (does it involve few or several elements)?
- **How frequently must the behaviour be performed?**

- **How compatible is the proposed behaviour** with the audience's current practices (is the behaviour socially approved)?
- **Are there major barriers** to engaging in the desired behaviour? What information, skills, resources and/or access must the audience segment acquires to overcome the barriers and make the desired behaviour change?
- **Are there at least some** members of the segment ("doers") **who manage to do the desired behaviour**? Do they have unusual characteristics?

CHAPTER -3
SYSTEM ANALYSIS AND DESIGN

SYSTEM ANALYSIS AND DESIGN

User of Digital Key of Digital India will have to upload their own document i.e., reference book in pdf format, book's content page in pdf format and finally the document of which he wants to create references. User will get her/his required information on the browser.

Thinking of End Users, application requirements such as the following may be generated.
The software, Web Based System, which is designed for administrating and automating all the major activities that are carried out by the users.

3.1 FUNCTIONAL REQUIREMENTS:

- Existing pdf (Book or journal to be uploaded) (Pdf1).
- A Pdf of the contents Page of Pdf1. (Pdf2)
- Any random pdf on topics related to Pdf1.(Pdf3)

3.2 NON-FUNCTIONAL REQUIREMENTS:

A Non-Functional requirement defines the quality attribute of a software system.
They represent a set of standards used to judge the specific operation of a system.
Examples—How fast does the website load?
There are few types of NFR.

- **Performance –** for example Response Time, Throughput, Utilization.
- **Scalability -** Scalability is the ability of the application to handle an increase in workload without performance degradation, or its ability to quickly enlarge
- **Availability -** how dependable the system is able to function during normal operating times

- **Reliability -** Reliability is usually defined as the probability that a product will operate without failure for a specified number of uses (transactions) or for a specified period of time.
- **Recoverability -** Recovery is the ability for a system to prepare and respond to a disaster.
- **Maintainability -** Maintainability is the ability of the application to go through changes with a fair degree of effortlessness.
- **Security -** Provider systems shall resist unauthorised, accidental or unintended usage and provide access only to legitimate users.
- **Interoperability -** Interoperability is the ability to exchange information and communicate with internal and external applications and systems.
- **Manageability -** Manageability is the ease with which the administrators can monitor the system, through critical health status exposed through its monitoring capabilities.

3.3 END USER REQUIREMENT

End user features include the following:

- Users should be able to use the **Digital Key of Digital India** from any Web browser/platform supporting HTML5 and cookies.
- Site visitors should be able to upload their pdfs easily.
- Users should be able to view a complete list of specified services available through the site.
- Users should be able to fulfil their requirements with very little efforts.
- The System must provide reliable results to the user.
- Large numbers of users should be able to use the application simultaneously.
- The performance of the application should not degrade with an increase in the number of services offered.

Modules of Digital Key of Digital :

- Folder creation Module: Creating a Folder which has the same name of the Book .
- Pdf to HTML converter Module: Used for converting for existing pdf to HTML using PdfDOMTree.
- Script Module: To add the JavaScript to the three converted HTML pages for Hyperlinking.

3.4 SOFTWARE REQUIREMENTS:

Visual Studio Code:

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). The source code is free and open source and released under the permissive MIT license.

SpringToolSuite4:

Spring Tools 4 is the next generation of Spring tooling for your favourite coding environment. Largely rebuilt from scratch, it provides world-class support for developing Spring-based enterprise applications, whether you prefer Eclipse, Visual Studio Code, or Theia IDE.

3.5 HARDWARE REQUIREMENTS:

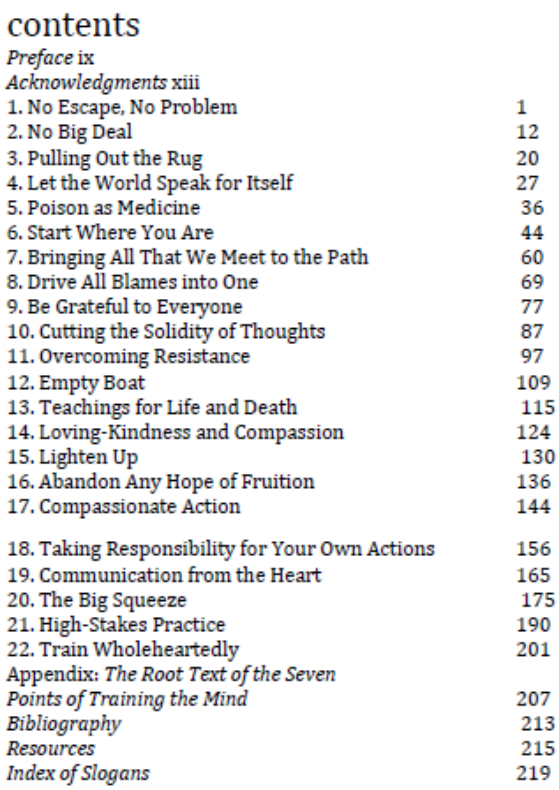
- Pentium/intel processor 233MHZ or above
- RAM Capacity 256MB
- Hard Disk Drive 20GB
- Operating System Windows 10
- Keyboard 108 Standard

3.6 DESIGN FEATURES:

- Interoperability
- Common Language Runtime Engine
- Language Independence
- Base Class Library
- Simplified Deployment
- Security
- Portability

3.7 DESIGN CONSTRAINTS:

- Size of Pdf being uploaded should not exceed 1.4 mb.
- Pdf-1 that is the book/journal to be converted into html, should have heading (mentioned in the contents page) in a different font-size compared to the text present
- Contents Page to be uploaded as Pdf-2 should be in the format specified in the image below



contents

Box in Red is the wrong way of Representation of Topic in Content Page, entire topic should be written in a single line as shown in Green Box.

CHAPTER -4

TOOLS AND TECHNOLOGIES

TOOLS AND TECHNOLOGIES

We have used multiple technologies, in front-end development we use HTML5, CSS3, Bootstrap and JavaScript and for Backend we have used Java. Following is the detailed information about these technologies:

4.1 FRONT END TECHNOLOGIES:

4.1.1 HTML5:

We use HTML5 as front-end for our project, **Digital Key of Digital India**. As HTML5 is very efficient for drag and drop like features.

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.

The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.

HTML5 is designed, as much as possible, to be backward compatible with existing web browsers. Its new features have been built on existing features and allow you to provide fallback content for older browsers.

It is suggested to detect support for individual HTML5 features using a few lines of JavaScript.

4.1.2 CSS3:

CSS stands for Cascading Style Sheets. CSS is a standard style sheet language used for describing the presentation (i.e., the layout and formatting) of the web pages.

Prior to CSS, nearly all of the presentational attributes of HTML documents were contained within the HTML mark-up (specifically inside the HTML tags); all the font colours, background styles, element alignments, borders and sizes had to be explicitly described within the HTML.

To solve this problem CSS was introduced in 1996 by the World Wide Web Consortium (W3C), which also maintains its standard. CSS was designed to enable the separation of presentation and content. Now web designers can move the formatting information of the web pages to a separate style sheet which results in considerably simpler HTML mark-up, and better maintainability.

CSS3 is the latest version of the CSS specification. CSS3 adds several new styling features and improvements to enhance the web presentation capabilities.

- You can easily apply same style rules on multiple elements.
- You can control the presentation of multiple pages of a website with a single style sheet.
- You can present the same page differently on different devices.
- You can style dynamic states of elements such as hover, focus, etc. that isn't possible otherwise.
- You can change the position of an element on a web page without changing the markup.

4.1.3 Bootstrap:

In the designing of home page of our project we use bootstrap as frontend designing. Login, register etc pages we use bootstrap because it is responsive and support almost every browser.

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Bootstrap is completely free to download and use. Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open-source product in August 2011 on GitHub.

- Bootstrap is a free front-end framework for faster and easier web development.
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins.
- Bootstrap also gives you the ability to easily create responsive designs.

4.1.4 JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g., functional programming) styles.

JavaScript has been used to run the browser for the following purposes:

- Pdf1.js – To read all the Headings and add unique id to each heading for Hyperlinking.
- Pdf2.js – To collect all the headings so that it can be matched with Pdf1 for Hyperlinking.
- Pdf3.js- Words in third pdf that are matching with headings of first pdf content will be hyperlinked to the same (clicking on the words will land the user to the specific part of the eBook i.e., Pdf1).

4.2 BACKEND TECHNOLOGIES:

4.2.1 Java

Java is a programming language and a platform. Java is a high level, robust, object-oriented and secure programming language. Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995. *James Gosling* is known as the father of Java. Before Java, its name was *Oak.* Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

Java Programming Language has been used for controlling the back-end part for the following:

- To convert from PDF to HTML using PdfDomTree library.
- To create a new Folder to store the converted pdfs into HTML.
- To add JavaScript to the Corresponding converted HTMLs for Hyperlinking.

Spring Boot -Java

Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications.

It is a Spring module that provides the RAD (*Rapid Application Development*) feature to the Spring Framework. It is used to create a stand-alone Spring-based application that you can just run because it needs minimal Spring configuration.

We ran our application using Tomcat Server which is in-built feature of Spring Boot. Spring Boot was further used as controller was used to handle the request from the user and send the appropriate Responses.

4.3 DATABASE:

One problem present is using System local storage. No other databases like MySQL, MongoDB, etc are used.

CHAPTER -5
SYSTEM IMPLEMENTATION

SYSTEM IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be completely new, replacing an existing manual or automated system or it may be major modification to an existing system.

5.1 DIRECT CONVERSION

This method converts from old to the new system abruptly, sometimes over a weekend or even overnight. The old system is used until a planned conversion day, when it is replaced by the new system. There are no parallel activities. The main disadvantage of this approach is: no other systems to fall back on, if difficulties arise with new system. Secondly, wise and careful planning is required.

5.2 PILOT SYSTEM

The Pilot approach is often preferred in the case of the new systems which involve new technique or some drastic changes in organization performance. In this method, a working version of the system is implemented in one part of the organization, such as single work area department. Based on feedback, the changes are made and the system is installed in the remaining departments of the organization, either all at once (direct conversion method) or (Gradually (phase-in method). This approach provides experience and lives test before implementation.

5.3 PHASE-IN METHOD

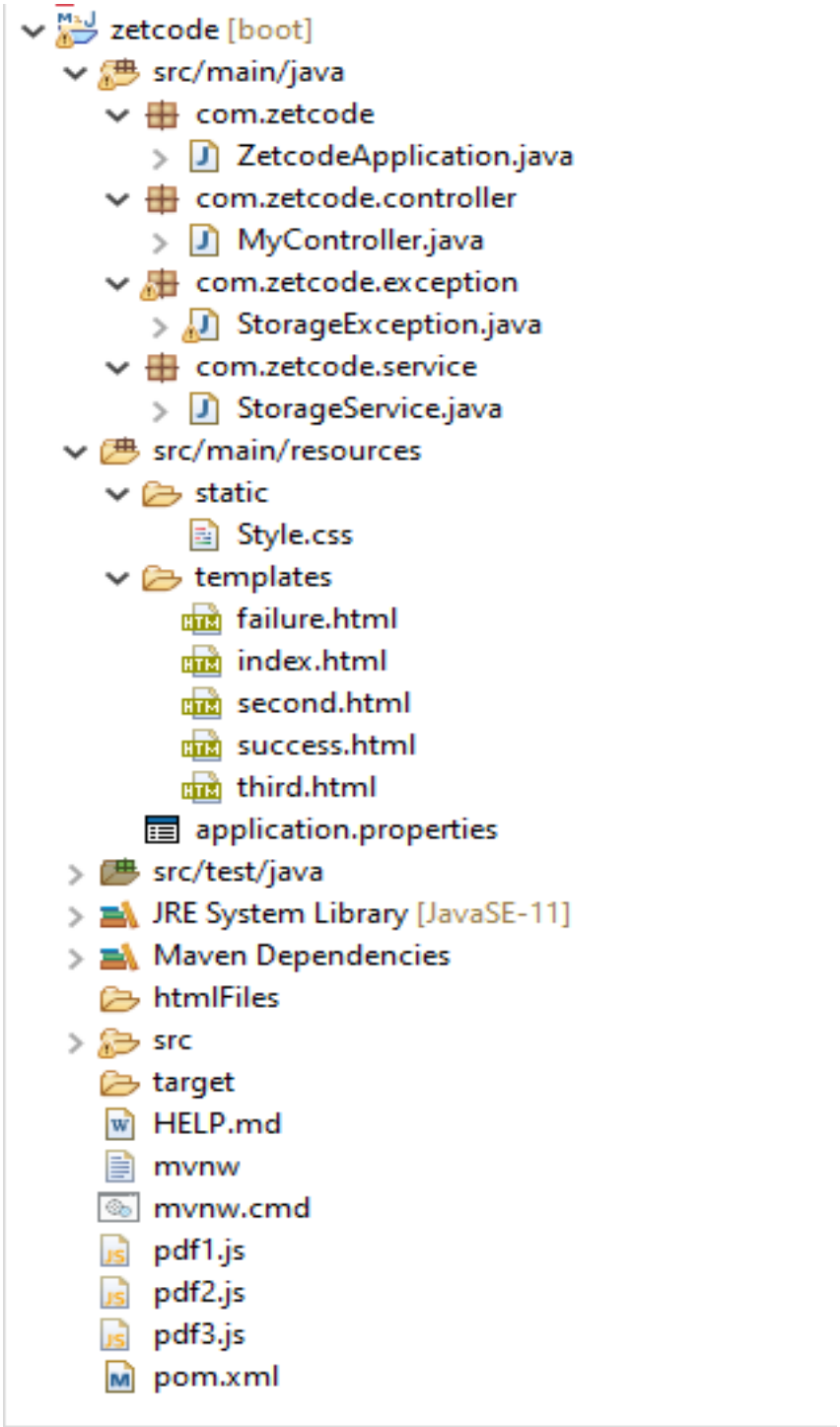
This method is used when it is not possible to install a new system through an organization all at once the conversion of files, training of the personnel or arrival of equipment may force the staging of the implementation over a period of time, ranging from weeks to months. It allows training and installation without unnecessary use of resources.

5.4 PARALLEL CONVERSION

If it is possible to keep the old system in place and running while the new system is installed, you gain certain benefits. We can directly compare the effectiveness and efficiency of the new and old systems. If the new system fails, the old system is still there chugging away as normal so no harm is done. Parallel conversion is only possible if the old and new systems are completely independent. It is impossible to achieve if the new system is built on the old system.

5.5 CODE:

Project Folder Structure:



POM.XML :-

The dependencies that are required in this project are stored in this pom.xml file.

```
   <?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
      <modelVersion>4.0.0</modelVersion>
      <parent>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-parent</artifactId>
            <version>2.5.4</version>
            <relativePath /> <!-- lookup parent from repository -->
      </parent>
      <groupId>com.mycode</groupId>
      <artifactId>mycode</artifactId>
      <version>0.0.1-SNAPSHOT</version>
      <name>zetcode</name>
      <description>Demo project for Spring Boot</description>
      <properties>
            <java.version>11</java.version>
      </properties>
      <dependencies>
            <dependency>
                  <groupId>org.springframework.boot</groupId>
                  <artifactId>spring-boot-starter-web</artifactId>
            </dependency>

            <dependency>
                  <groupId>org.springframework.boot</groupId>
                  <artifactId>spring-boot-starter-test</artifactId>
                  <scope>test</scope>
            </dependency>

    <dependency>
    <groupId>org.apache.pdfbox</groupId>
    <artifactId>pdfbox-tools</artifactId>
    <version>2.0.3</version>
</dependency>
<dependency>
    <groupId>net.sf.cssbox</groupId>
    <artifactId>pdf2dom</artifactId>
    <version>2.0.0</version>
</dependency>

<dependency>
    <groupId>com.itextpdf</groupId>
    <artifactId>itextpdf</artifactId>
    <version>5.5.10</version>
```

```
</dependency>
<dependency>
    <groupId>com.itextpdf.tool</groupId>
    <artifactId>xmlworker</artifactId>
    <version>5.5.10</version>
</dependency>



<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.15</version>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-scratchpad</artifactId>
    <version>3.15</version>
</dependency>
      <dependency>
                  <groupId>org.apache.pdfbox</groupId>
                  <artifactId>pdfbox</artifactId>
                  <version>2.0.22</version>
            </dependency>
            <dependency>
                  <groupId>net.mabboud.fontverter</groupId>
                  <artifactId>FontVerter</artifactId>
                  <version>1.2.22</version>
            </dependency>
            <dependency>
                  <groupId>commons-io</groupId>
                  <artifactId>commons-io</artifactId>
                  <version>2.8.0</version>
            </dependency>


            <dependency>
                  <groupId>org.jsoup</groupId>
                  <artifactId>jsoup</artifactId>
                  <version>1.13.1</version>
                  <scope>test</scope>
            </dependency>
            <dependency>
                  <groupId>org.hamcrest</groupId>
                  <artifactId>hamcrest-all</artifactId>
                  <version>1.3</version>
                  <scope>test</scope>
            </dependency>


            <dependency>
                  <groupId>net.mabboud.gfxassert</groupId>
                  <artifactId>GfxAssert</artifactId>
                  <version>1.0.4</version>
```

```
			<scope>test</scope>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-thymeleaf</artifactId>
		</dependency>


	</dependencies>


	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>

</project>
```

1 My Controller :-

My controller read the PDF and saves it to the chosen directory .

Uploading File to Folder :-

```
import com.zetcode.exception.StorageException;
import com.zetcode.service.StorageService;

import java.io.File;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
//import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.MultipartFile;

@Controller
public class MyController {

    @Autowired
    private StorageService storageService;

    @RequestMapping(value = "/doUpload", method = RequestMethod.POST,
```

```
            consumes = {"multipart/form-data"})
    public String upload(@RequestParam MultipartFile file) {

        storageService.uploadFile(file);

        return "second.html";
    }

    @ExceptionHandler(StorageException.class)
    public String handleStorageFileNotFound(StorageException e) {

        return "redirect:/failure.html";
    }



    //2nd file uploading
    @RequestMapping(value = "/dodoUpload",
            method = RequestMethod.POST,
            consumes = {"multipart/form-data"})
    public String seconduplоad(@RequestParam MultipartFile file) {
System.out.println("this is second");
        storageService.upload2File(file);

        return "third.html";
    }

    //3nd file uploading
    @RequestMapping(value="/success",
            method = RequestMethod.POST,
            consumes = {"multipart/form-data"})
    public String thirdupload(@RequestParam MultipartFile file) {
System.out.println("this is second");
        storageService.upload3File(file);

        return "redirect:/file";
    }


    @RequestMapping(value = "/file", method = RequestMethod.GET)
      public String showFile() {
            // Creating a File object for directory
            File directoryPath = new
File("C:\\Users\\NIKHIL\\Desktop\\Practice\\Hibernateworkspace\\New\\zetcode
\\htmlFiles");
            // List of all files and directories
            String[] temp = directoryPath.list();
            for (int i = 0; i < 3; i++) {
                  String contents = (String)
("C:\\Users\\NIKHIL\\Desktop\\Practice\\Hibernateworkspace\\New\\zetcode\\ht
mlFiles\\" + temp[i].toString());

                  try {
                        Runtime rTime = Runtime.getRuntime();
```

```
                        String url = contents;
                        String browser = "C:\\Program
Files\\Google\\Chrome\\Application/chrome.exe ";
                        Process pc = rTime.exec(browser + url);
                        pc.waitFor();

                  } catch (Exception e) {
                        System.out.println(e);
                  }
            }
            return "redirect:/success.html";
      }
}
```

2.We have a handler for the `StorageException`.

```
package com.zetcode.service;

import com.zetcode.exception.StorageException;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;

@Service
public class StorageService {

    @Value("${upload.path}")
```

```
    private String path;

    public void uploadFile(MultipartFile file) {

        if (file.isEmpty()) {

            throw new StorageException("Failed to store empty file");
        }
        try {
            var fileName = file.getOriginalFilename();
            var is = file.getInputStream();

            Files.copy(is, Paths.get(path + fileName),
                    StandardCopyOption.REPLACE_EXISTING);
        } catch (IOException e) {

            var msg = String.format("Failed to store file %f", file.getName());

            throw new StorageException(msg, e);
        }
    }
}
```

This is our custom `StorageException`. It is thrown when the file cannot be stored on the filesystem.

3 .`StorageService` copies the PDF from the inputstream, convert it to HTML, add the <script> tag for Hyperlinking and saves it on the disk.

```
import com.zetcode.exception.StorageException;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;
import org.springframework.web.multipart.MultipartFile;
```

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;

//pdftoHtml dependencies
import java.io.File;
import java.io.PrintWriter;
import java.io.Writer;

import org.apache.pdfbox.pdmodel.PDDocument;
import org.fit.pdfdom.PDFDomTree;

//HTMLmanupulation
import java.io.BufferedWriter;

import java.io.FileWriter;


@Service
public class StorageService {

    @Value("${upload.path}")
    private String path;
  private String
opath="C:\\Users\\NIKHIL\\Desktop\\Practice\\Hibernateworkspace\\New\\zetcod
e\\htmlFiles\\";
    public void uploadFile(MultipartFile file) {
      System.out.println("uploading");

        if (file.isEmpty()) {

            throw new StorageException("Failed to store empty file");
        }

        try {
            var fileName = file.getOriginalFilename();
            var is = file.getInputStream();


            Files.copy(is, Paths.get(path + fileName),
                    StandardCopyOption.REPLACE_EXISTING);
            System.out.println("File Uploaded");

            //mycode
            generateHTMLFromPDF(path + fileName,"pdf1");
             System.out.println("File Converted");
             System.out.println(opath);
             //HTMLmanupulation
            String pat=opath + "pdf1"+".html";
            htmlManupulation(pat);
            System.out.println("Script tag added");
        } catch (IOException e) {
```

```
            var msg = String.format("Failed to store file %f",
file.getName());

            throw new StorageException(msg, e);
        }


    }

    public void upload2File(MultipartFile file) {
      System.out.println("uploading");

        if (file.isEmpty()) {

            throw new StorageException("Failed to store empty file");
        }

        try {
            var fileName = file.getOriginalFilename();
            var is = file.getInputStream();

            Files.copy(is, Paths.get(path + fileName),
                    StandardCopyOption.REPLACE_EXISTING);
            System.out.println("File Uploaded");

            //mycode
            generateHTMLFromPDF(path + fileName,"pdf2");
             System.out.println("File Converted");

             //HTMLmanupulation
            String pat=opath + "pdf2"+".html";
            htmlManupulation2(pat);
            System.out.println("Script tag added");
        } catch (IOException e) {

            var msg = String.format("Failed to store file %f",
file.getName());

            throw new StorageException(msg, e);
        }


    }

    public void upload3File(MultipartFile file) {
      System.out.println("uploading");

        if (file.isEmpty()) {

            throw new StorageException("Failed to store empty file");
        }

        try {
```

```
            var fileName = file.getOriginalFilename();
            var is = file.getInputStream();

            Files.copy(is, Paths.get(path + fileName),
                    StandardCopyOption.REPLACE_EXISTING);
            System.out.println("File Uploaded");

            //mycode
            generateHTMLFromPDF(path + fileName,"pdf3");
             System.out.println("File Converted");

             //HTMLmanupulation
            String pat=opath + "pdf3"+".html";
            htmlManupulation3(pat);
            System.out.println("Script tag added");

            //display

        } catch (IOException e) {

            var msg = String.format("Failed to store file %f",
file.getName());

            throw new StorageException(msg, e);
        }


//Converting PDF to HTML
    }

    private  void generateHTMLFromPDF(String filename,String fname) {
      try {
          PDDocument pdf = PDDocument.load(new File(filename));
          Writer output = new PrintWriter(opath + fname+".html");
          new PDFDomTree().writeText(pdf, output);

          output.close();
      }
       catch(Throwable e) {
            e.printStackTrace();
}
//Adding Script tag in html files.
}
public void htmlManupulation(String pat) throws IOException {

            File f=new File(pat);//place where html is there

            FileWriter fw = new FileWriter(f, true);
            BufferedWriter bw=new BufferedWriter(fw);
            String s="<script src='pdf1.js'></script>";
            bw.write(s);//will include .js file in html
            bw.close();

      }
```

```
public void htmlManupulation2(String pat) throws IOException {

	File f=new File(pat);//place where html is there

	FileWriter fw = new FileWriter(f, true);
	BufferedWriter bw=new BufferedWriter(fw);
	String s="<script src='pdf2.js'></script>";
	bw.write(s);//will include .js file in html
	bw.close();

}
public void htmlManupulation3(String pat) throws IOException {

	File f=new File(pat);//place where html is there

	FileWriter fw = new FileWriter(f, true);
	BufferedWriter bw=new BufferedWriter(fw);
	String s="<script src='pdf3.js'></script>";
	bw.write(s);//will include .js file in html
	bw.close();

}



}
```

This is the home page. It is a static file located in the `src/main/resources/static` directory. It contains a form to select a file and send it to the Spring application

Index.html page

```
<!DOCTYPE html>

<html lang="en">


  <head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>
```

```
<style>
  * {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
  }

  html {
    font-family: "Montserrat", sans-serif;
  }

  body {
    background: #eee;
    padding: 0 0px;
  }

  header {
    background-color: #f5ba13;
    padding: 21px 32px;
    box-shadow: 0 0 10px 0 rgb(0 0 0 / 30%);
    height: 100px;
    font-size: 24px;
  }

  header h1 {
    color: #fff;
    font-family: Arial, Helvetica, sans-serif;
    font-weight: 225;
    text-align: center;
  }

  .navbar {
    display: flex;
    align-items: center;
```

```
  position: stickey;
  height: 80px;
  background-color: #f5ba13;
  justify-content: space-between;
}

.navlist {
  display: flex;
  align-items: center;
  list-style: none;
}

.navlist a {
  text-decoration: none;
  color: white;
  font-size: 23px;
  padding-left: 87px;
}

.navlist li a:hover {
  color: grey;
}

.rightNav {
  display: flex;
  align-items: center;
}

.rightNav input {
  padding: 5px;
  height: 25px;
  font-size: 15px;
  border: 2px solid grey;
  border-radius: 11px;
```

```
  outline: 0px;
  width: 180px;
}

.rightNav button {
  background-color: orange;
  margin: 10px;
  height: 23px;
  width: 28px;
  text-transform: inherit;
}

.upload {
  display: flex;
  justify-content: center;
  margin-top: 171px;
  margin-bottom: 30px;
}

#btn {
  height: 37px;
  width: 85px;
  background-color: orange;
  color: black;
  border: 4px solid grey;
  border-radius: 7px;
  font-size: 17px;
}

label {
  font-size: 26px;
}

input {
```

```
      font-size: 18px;
      height: 27px;
      width: 343px;
    }

    h1 {
      margin-top: 20px;
      text-align: center;
    }

    footer {
      position: fixed;
      text-align: center;
      bottom: 0px;
      width: 100%;
      height: 2.5rem;
      margin-top: 7px
    }

    footer p {
      color: orange;
    }

    .contraints {
      text-align: center;
    }
  </style>
</head>

<body>

  <!-- Header Section -->
  <Header>
    <h1>Digital Key of Digital India</h1>
```

```
    </Header>

    <!-- Navbar Section -->
    <div class="navbar">
      <div class="leftNav">
        <ul class="navlist">
          <li>
            <a href="/home">Home</a>
          </li>
          <li>
            <a href="/signup">SignUp</a>
          </li>
          <li>
            <a href="/upload">Upload</a>
          </li>
        </ul>
      </div>
    </div>

    <!-- Input Section -->
    <form class="upload" action="/doUpload" enctype="multipart/form-data">
      <label>Select files:</label>
      <input type="file" name="file" multiple accept=".pdf,.docx" />
      <button id="btn">Submit</button>
    </form>

    <div class="contraints">
      <h3>Constraints</h3><br>
      <ol>
        In this application file maximum 1 mb file can be uploaded
      </ol>
    </div>

    <footer>
```

```
      <p>
        Copyright &copy;2021 Cdac.in
      </p>
    </footer>
  </body>

</html>
```

We have chosen the `doUpload` URL pattern. A request created by this form will be processed by a Spring controller. The `enctype` attribute specifies the `multipart/form-data` encoding type, which is required for uploading a file with an HTML form.

The `success.html` is shown when the file is successfully uploaded to the server.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Success</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <p>File successfully uploaded</p>
    </body>
</html>
```

Front-end Logics:

PDF1.js:

Pdf2.js is a JavaScript file which will add id to the headings of the pdf1 i.e., the book, so that they can be hyperlinked.

Explanation along with the Code of pdf1.js:

```
//to get all Font Size
 let toGetAllFont=[];
let y=0;//index for above loop
let words=document.getElementsByTagName("div");

addingAllFont();

function addingAllFont(){

 for(let i=0;i<words.length;i++){

   const getFont=words[i].style.getPropertyValue("font-size");
   //console.log("printing font"+getFont);
   if(toGetAllFont.length==0){
      toGetAllFont[y]=getFont;
      y++;
   }

   else {
      let x=true;
      for(let z=0;z<toGetAllFont.length;z++){
         if(toGetAllFont[z]===getFont){
```

```
            x=false;
          }

      }
      if(Boolean(x)){
         toGetAllFont[y]=getFont;
         y++;
      }
   }
 }
}

for(let io=0;io<toGetAllFont.length;io++){

   console.log(io+" coming here "+toGetAllFont[io]);//displaying all the font Size
}

// end of logic to get all font -size

var importedData=localStorage.getItem('VariableContainData');//importing from cache

console.log(importedData);

var string_to_array = function (str1) {
   return str1.trim().split("+");//Getting each line sepeartely
};

var data=string_to_array(importedData);

//console.log(data);
```

```
var filteredArr = data.filter(function (elem) {
   return elem !== undefined;
 });

 let withReg=[];
 for(i in filteredArr)
 {
   withReg[i]=filteredArr[i].replace(/[0-9.]/g,'');

 }//getting only the lines without page numbers

 for(let xy=0;xy<withReg.length;xy++){
   console.log(withReg[xy].trim());
   addHyperLink(withReg[xy].trim());
 }//removing additional Spaces if any



 function addHyperLink(eachString){//function to add hyperPlink to each Striing passed

   let allContents=eachString;

let splitContent=allContents.split(" ");

for(i=0;i<words.length;i++){

   if(words[i].style.getPropertyValue("font-
size")===toGetAllFont[2] || words[i].style.getPropertyValue("font-size")===toGetAllFont[1]

    || words[i].style.getPropertyValue("font-size")===toGetAllFont[4]
   || words[i].style.getPropertyValue("font-size")===toGetAllFont[5]){
```

```
        if(words[i].textContent===splitContent[0]){
          // console.log(words[i]);
           //console.log("Matching with World");
           //let count=1;
           var checkString="";
           for(let r=0 ;r<splitContent.length;r++){
              //console.log("Coming Inside loop");
              //console.log(words[i+r].textContent+"printing");
              //console.log(splitContent[r]+" text-content");
              if(r!=splitContent.length-1){

              if(splitContent[r]===words[i+r].textContent){
                 //count++;
                 //console.log(count);
                 //console.log("inside if"+splitContent[r]);
                 checkString+=splitContent[r]+" ";

              }
             }
             else{
              if(splitContent[r]===words[i+r].textContent || splitContent[r]===words[i+r+1].textContent ||
splitContent[r]===words[i+r+2].textContent){
                 //count++;
                 //console.log(count);
                // console.log("inside if"+splitContent[r]);
                 checkString+=splitContent[r]+" ";

             }
           }
           //console.log(checkString+" this is check string");
```

```
        if(checkString===(allContents+" ")){//if the heading matches with the string sent
                                    //from pdf2 only then Unique Id is added
            splitContent.reverse();
            let id=splitContent[0]+101;
            words[i].setAttribute("id",id);
            console.log(words[i]);
            console.log("added ");
          }
       }
    }
}
}

 }//end of function
```

PDF2.js:

Pdf2.js is a JavaScript file which will be added to pdf2.html file(containing headings/content page of book or pdf1).

The logics given in the script will pick the headings of the book that is given in pdf2 and store it in local cache memory so that it can be retrieved and used by pdf3.js and pdf1.js file for hyperlinking purpose.

```
var wordsOfPage=document.getElementsByTagName("div"); //getting elements in div tags.
 //Conversion with this library i.e., PDFDomTree gives the html page with each word in a div tag.

  var wordsline="";

  var y=0;

  //adding unique class names to each line words on the basis of their position i.e. top.
  for(var i=1;i<wordsOfPage.length;i++)
  {


  if(i===wordsOfPage.length-1)
  { //last word of the content
    y++;
    className="line";
    className=className+y; //eg. line1,line2....
    wordsOfPage[i].classList.add(className);
      break;
  }
  else if(((wordsOfPage[i].style.getPropertyValue("top") != wordsOfPage[i-1].style.getPropertyValue("top")) && (wordsOfPage[i].style.getPropertyValue("top") != wordsOfPage[i+1].style.getPropertyValue("top"))) == true)
    { //logic for differentiating last word of a line and first word of next line.

    y++;
    className="line";
    className=className+y;
    wordsOfPage[i].classList.add(className); //adding class names i.e., for words in line no 1 will have class name=line1.

    }
```

```
  else if(wordsOfPage[i-1].style.getPropertyValue("top") ===
wordsOfPage[i].style.getPropertyValue("top") )
  {// logic to add classes to words which are in between first and last word of the line.

     wordsOfPage[i-1].classList.add(className);
     wordsOfPage[i].classList.add(className);


    }
    else{
      y++;
      className="line";
      className=className+y; //line0, line1....

    }
  }

  var count=y; //count contains number of lines

    var abcval=" ";
    var countLine=1;
  for(var u=0;u<=count;u++) //logic for collecting all the words of a same line and putting it to a
strings
  {


    var classofLine="line"+countLine;
    var abc=document.getElementsByClassName(classofLine);



   for(var z=0;z<abc.length;z++)//logic to collect
   {
     console.log(z);
     abcval+=abc[z].textContent+" ";
   }

    abcval+="+"; //add '+' after each lin so that it can be splitted by split function to get back the
lines

    countLine++;


  }
```

```
  //Storing the string in cache memory so that it can be accessed and used from pdf1.html and
pdf2.html
  localStorage.setItem("VariableContainData",abcval);



//end of pdf2.js
```

PDF3.js

Pdf3.js is JavaScript file which is to be added to pdf3.html file and it will hyperlink the matching words of pdf3 to pdf1(headings).

```
let words=document.getElementsByTagName("div");// Reading all the Words in the Document

var importedData=localStorage.getItem('VariableContainData');//Importing the words from cache of//
Contents Page

console.log("Working");

console.log(importedData);


var string_to_array = function (str1) {

   return str1.trim().split("+");

};//Sepearating the each Line


var data=string_to_array(importedData);


//console.log(data);


var filteredArr = data.filter(function (elem) {

   return elem !== undefined;

 });
```

```
let withReg=[];
for(i in filteredArr)
{
  withReg[i]=filteredArr[i].replace(/[0-9]/g,'');

}


for(let xy=0;xy<withReg.length;xy++){
  console.log(withReg[xy].trim());
  addHyperLinkToeachWOrd(withReg[xy].trim());
}

//to add HyperLink to every encountered word
function addHyperLinkToeachWOrd(string1){
let allContents=string1;

let splitContent=allContents.split(" ");
for(i=0;i<words.length;i++){
  if(words[i].textContent===splitContent[0]){

    var checkString="";
      for(let r=0 ;r<splitContent.length;r++){
        //console.log("Coming Inside loop");
        console.log(words[i+r].textContent+"printing");
        console.log(splitContent[r]+" text-content");


        if(splitContent[r]===words[i+r].textContent){
          //count++;
          //console.log(count);
```

```
            console.log("inside if"+splitContent[r]);
            checkString+=splitContent[r]+" ";

          }


        }
        console.log(checkString+" this is check string");

        if(checkString===(allContents+" ")){//adding the Hyperlink if the entire String matches

          splitContent.reverse();

          var link='bookm.html#'+splitContent[0]+101;

          for(let r=0 ;r<splitContent.length;r++){
          let text=words[i+r].textContent;

         words[i+r].innerHTML='<a  href="'+link +'">' + text+ '</a>';
          }
        }
      }
 }
}
```

Index.html:

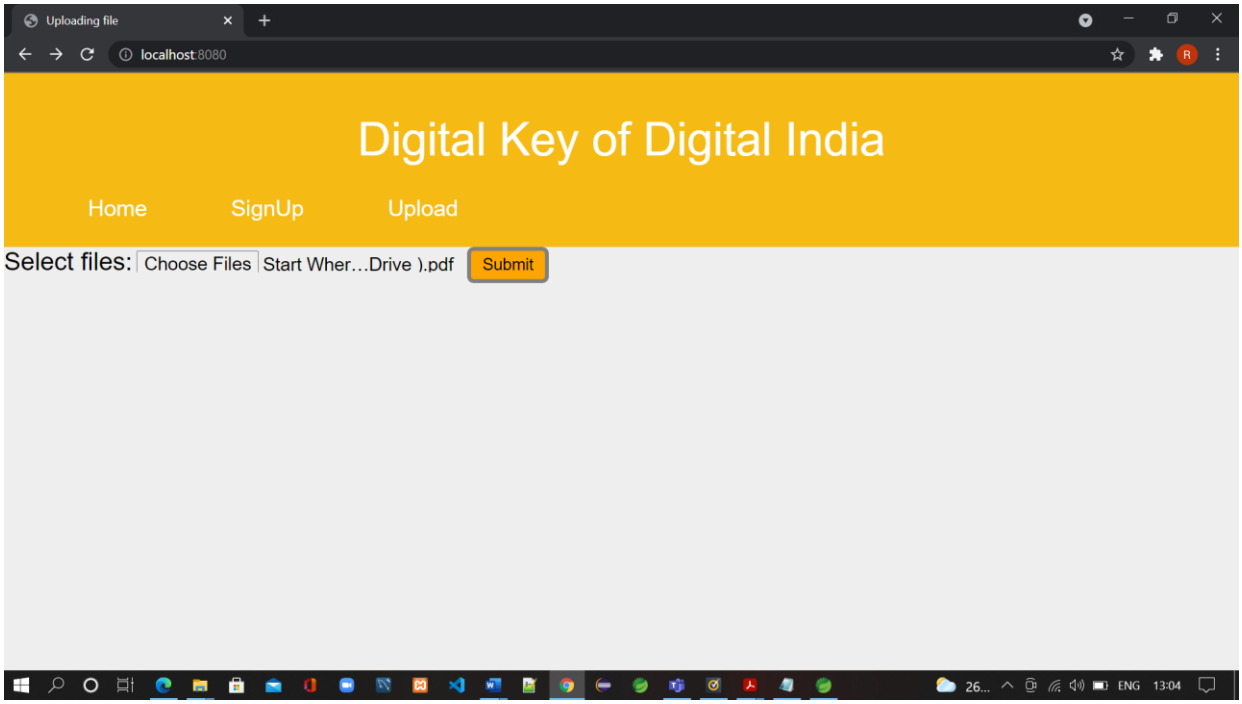


Fig. No. 1.3

Second.html:

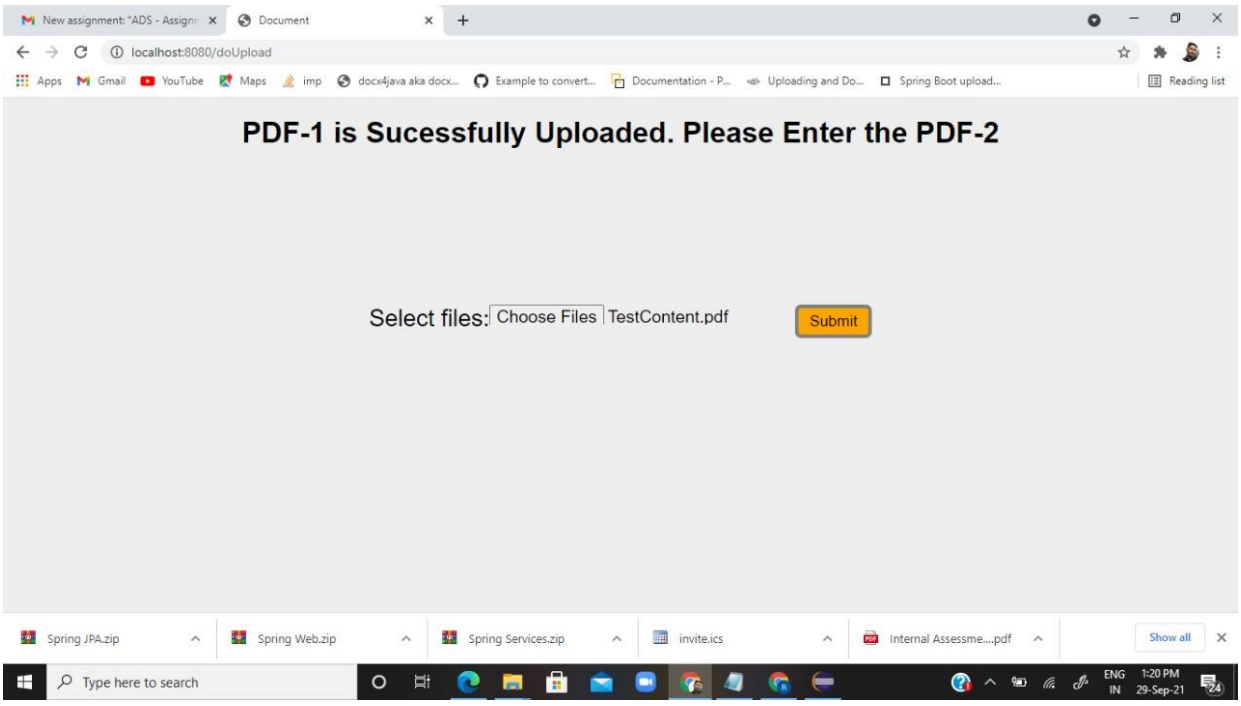


Fig. No. 1.4

Third.html:

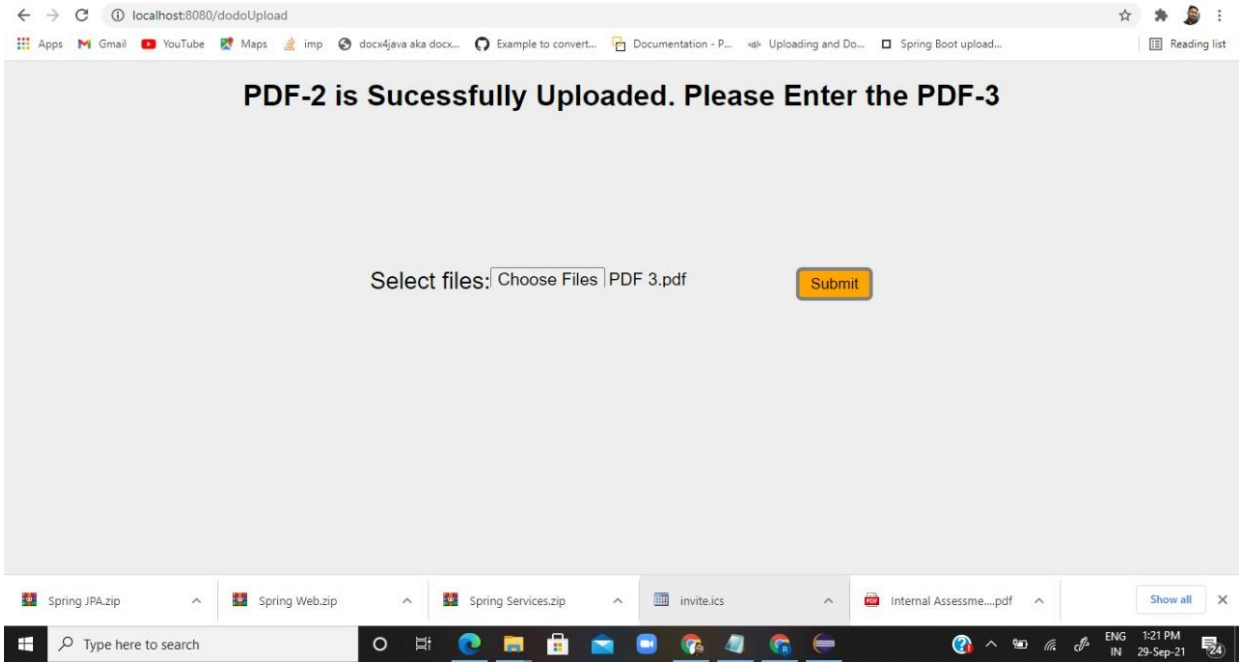


Fig. No. 1.5

Failure.html:

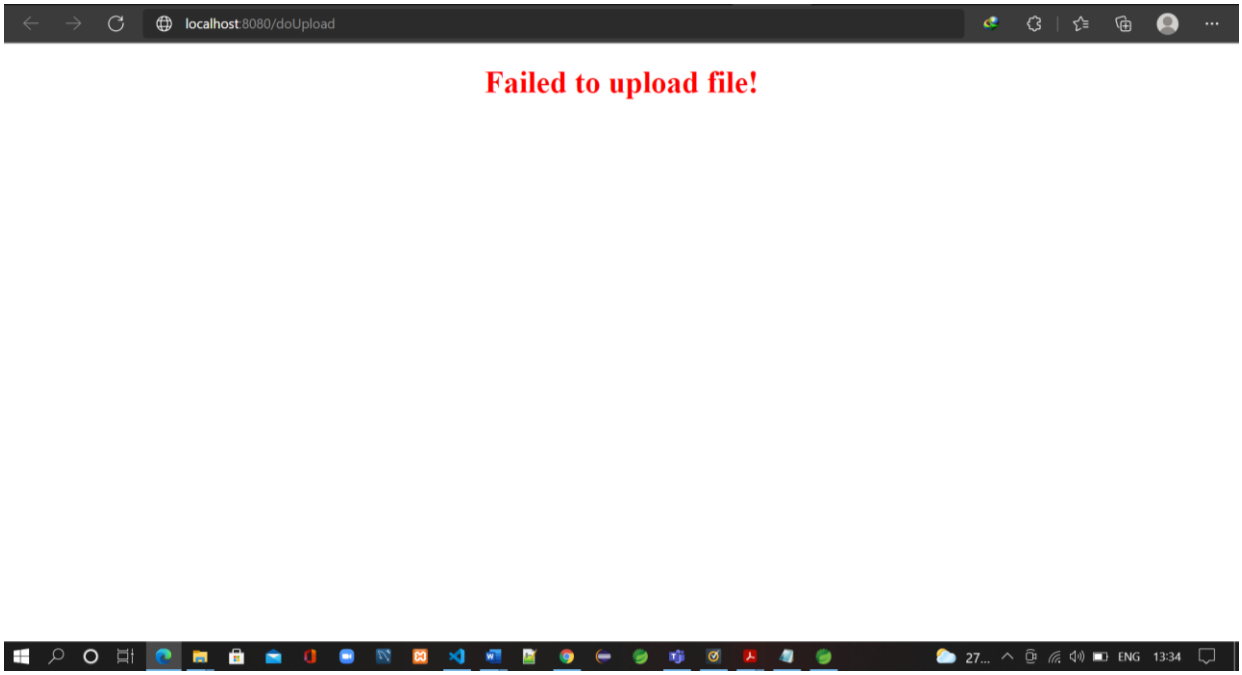


Fig. No. 1.6

5.6 SYSTEM MAINTENANCE

It needs to continually maintain the system. Programmers/ analyst spend sufficient time for maintaining programs.

The study on the maintenance requirement for the information system revealed that:

- 60-90 percent of the overall cost of software during the life of the system is spent on the maintenance.
- In document cases, the cost of maintenance, when measured on the basis of writing each instruction in coding form, is more than 50 times the cost of developing a system.
- The software demand is increasing at faster rate than supply, there is a backlog of new development work.

The maintenance can be classified as corrective, adaptation or perfective Corrective maintenance means repairing, processing or performance failure or making alterations because of previous ill-defined problems Adaptation maintenance means changing the program function. Enhancing performance, the programs according to user's additional needs are included in perfective maintenance more time and money are spent on perfective than on corrective and adaptive maintenance together Maintenance covers a wide range of activities including correcting coding and design errors, updating documentation and test data and upgrading user support.

CHAPTER-6
RESULT AND DISCUSSION

RESULT AND DISCUSSION

As a result, this web application will help the user in searching required information in a very effect way and with very less effort .

6.1 FUTURE SCOPE

This project has great potential to shape the future of the human kind. More and more features can be added to this project to make is more efficient and beneficial.
Some of the features which will be added are as follows:-

- Create a centralized server where large numbers of books can be stored which will serve as the references.
- Function for user to create her/his own repository where he can store her/his pdf books and get references of it.
- Career as well as job opportunities.

6.2 CONCLUSION

The project, **Digital Key of Digital India** have a tremendous scope in near future. It is an initiative towards digitalizing of document in such a way that it that they can be accessed and navigated easily. Since the concept used in the project is in the nascent state so we have encountered constraints and used assumption to realise this project. But we are sure in that this project can be restructured with addition of new findings which will increase its effectiveness.

At last, but not the least we think this concept and the project has a futuristic approach.

CHAPTER -7
BIBILIOGRAPHY

BIBILIOGRAPHY

[1] https://stackoverflow.com

[2] http://www.w3schools.com

[3] https://www.javatpoint.com/

[4] https://spring.io/

[5] https://www.javascript.com

[6] https://github.com/radkovo/Pdf2Dom/blob/master/src/main/java/org/fit/pdfdom/PDFDomTree.java

[7] http://cssbox.sourceforge.net/pdf2dom/documentation.php