AUDICODE

A

PROJECT REPORT

Submitted in partial fulfillment of the requirement for the award of Degree of

BACHELOR OF COMPUTER APPLICATIONS

Submitted to

CAREER POINT UNIVERSITY

ALANIYA, KOTA (Raj.)

PROJECT GUIDE:
Ms. Akshita Bhatnagar (Internal)

SUBMITTED BY:
Harshit Suman (K22132)

Major Project Report (CSD-675)
School of Computer Applications
CAREER POINT UNIVERSITY, ALANIYA, KOTA
Session (23-24)

ACKNOWLEDGEMENT

First of all, I would like to express my heartfelt gratitude to School of Computer application, Career Point University, Kota, Rajasthan for giving me this precious opportunity to follow a world recognized degree program.

My heartfelt gratitude goes to, Dr. Garima Tyagi ma'am (HOD of School of Computer Application), Mr. Arshad Hussain, Dr. Abid Hussain sir, Dr. Amit Sharma sir, Ms. Shalini Chawla ma'am, Ms. Akshita Bhatnagar, Mr. Praveen Goyal of school of Computer application at Career Point University for their valuable suggestions and directly or indirectly for completing this project.

Finally, I would like to express my special thanks to All my Classmates for wisdom Computer Technologies and staff members for giving me the great support and for providing their valuable time to help me for successful completing This project.

Thanks goes to all those who helped, whether through their comments, feedback, edits or suggestion.

DECLARATION

We hereby declare that this Project Report titled AudiCode submitted by us and approved by our project guide, the School of Computer Application and Technology (SOCA), Career Point University, Kota is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Project Name :	AudiCode	
Student Name:	Harshit Suman	Signature
Project Guide: (Internal)	Ms. Akshita Bhatnagar	Signature

Table of Contents

Appendix

1 Project Title

"AudiCode: PDF to Audiobook Converter and QR Code Generator"

The AudiCode project aims to provide a comprehensive solution for converting PDF documents into audiobooks and generating QR codes. In today's digital age, access to information in various formats is crucial for enhancing accessibility and convenience. With AudiCode, users can seamlessly convert text-based PDFs into audio files, making content more accessible to visually impaired individuals and those on the go. Additionally, the project offers a QR code generation feature, allowing users to create QR codes for links, text, videos, audio files, and more. By combining these functionalities into a single platform, AudiCode simplifies the process of accessing and sharing information across different mediums.

2 Problem Statement

- **Accessibility Challenge:** Individuals with visual impairments or reading difficulties find it hard to consume text-heavy content in PDFs.

- **Time Consumption:** Reading lengthy PDF documents can be time-consuming and inconvenient for many users.

- **Auditory Learning:** There is a growing preference for auditory learning and content consumption, which is not supported by traditional text documents.

- **Manual QR Code Generation:** Generating QR codes manually for different types of content (links, videos, audio, photos) is cumbersome and time-consuming.

- **Integration Gap:** There is a lack of integrated tools that combine text-to-audio conversion and QR code generation in a single, user-friendly platform.

- **Technical Barriers:** Non-technical users face difficulties in using complex software or tools for converting PDFs to audiobooks and generating QR codes.

- **Cost Barrier:** Many available solutions for text-to-speech and QR code generation are paid, limiting access for users looking for free or affordable options.

3 Project Description

AudiCode is a web application designed to convert PDF documents into audiobooks and generate QR codes for various types of content, including links, videos, audio, and photos. The application aims to provide a user-friendly interface for uploading PDF files, extracting text, converting the text into speech, and generating downloadable audio files. Additionally, it allows users to create QR codes from input data, which can be displayed and downloaded.

3.1 Scope of the Work

The project scope includes:

- Developing a web application with a responsive and user-friendly interface.
- Implementing PDF upload functionality and text extraction.
- Converting extracted text into an audio format.
- Enabling users to generate and download QR codes.
- Hosting the application on a free platform to ensure accessibility.
- Ensuring the application works across different devices and browsers.

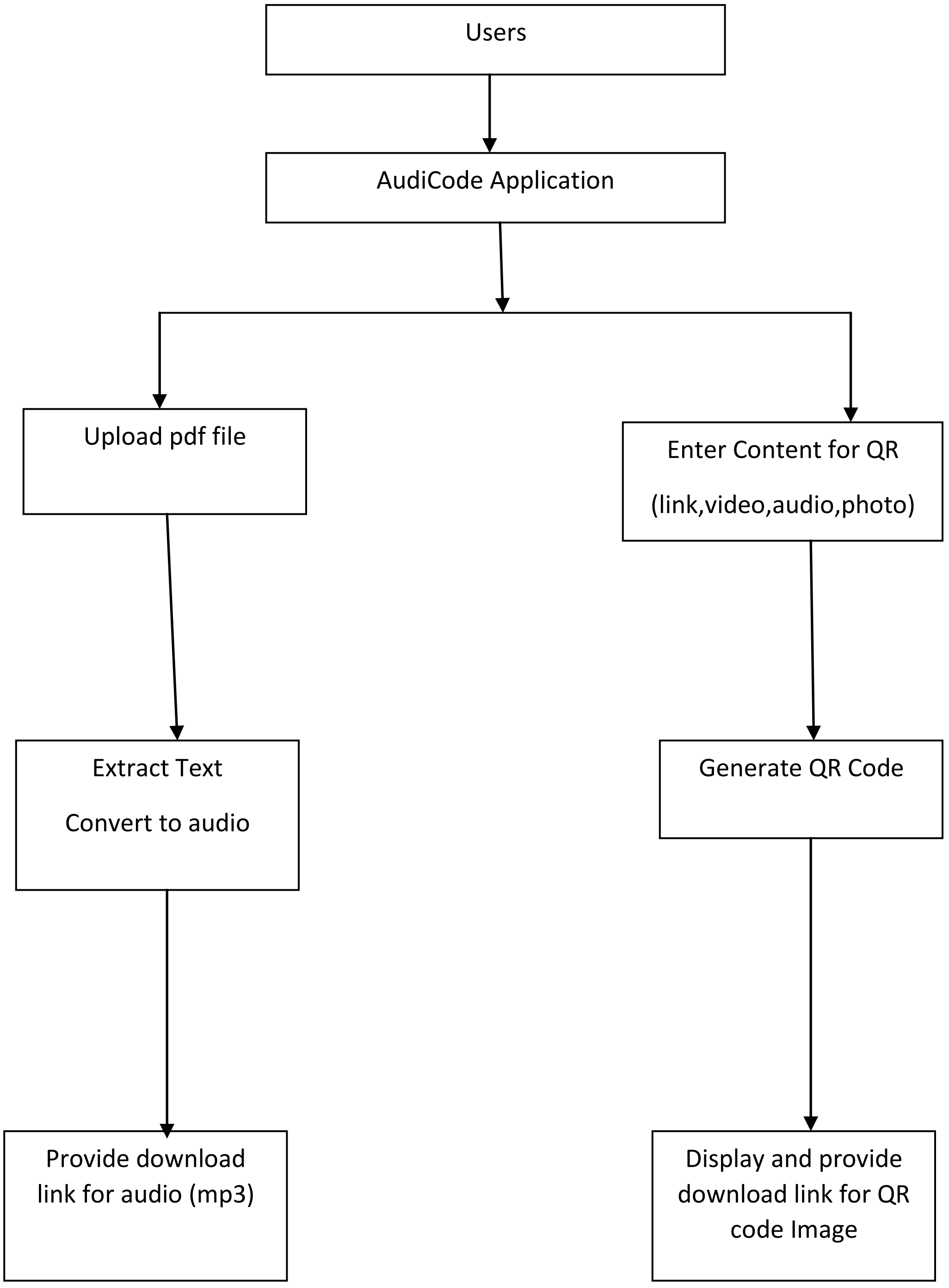
3.2 Project Modules

Home Module: Selection screen for converting PDFs to audiobooks or generating QR codes.

PDF to Audiobook Module: Uploading PDFs, extracting text, converting text to audio, and downloading the audio file.

QR Code Generator Module: Input form for generating QR codes for different content types (link, video, audio, photo), displaying the QR code, and providing a download option.

3.3 Context Diagram (High Level)



4 Implementation Methodology

1. Planning

Requirement Analysis: Identify and document project requirements.

Scope Definition: Outline the project features and limitations.

Resource Allocation: Assign necessary resources and tools.

2. Design

System Architecture: Create high-level diagrams (context and data flow diagrams).

UI/UX Design: Design wireframes and mockups for a responsive interface.

Database Design: (If needed) Design the schema for storing user data.

3. Development

Environment Setup: Configure the development environment with required tools.

Frontend Development: Develop the user interface using HTML, CSS, and JavaScript.

Backend Development: Implement backend logic using Flask.

Integration: Ensure seamless communication between frontend and backend.

4. Testing

Unit Testing: Test individual components and functions.

Integration Testing: Verify the integration between frontend and backend.

User Acceptance Testing (UAT): Gather feedback from potential users.

Bug Fixing: Address bugs identified during testing.

5. Deployment

Select Hosting Platform: Choose a free hosting service (GitHub Pages, Heroku, Render, or Pythonanywhere).

Prepare Deployment Scripts: Create necessary deployment files and scripts.

Deploy Application: Deploy to the selected platform.

Post-Deployment Testing: Test the application in the production environment.

6. Maintenance

Monitor Application: Continuously monitor performance and errors.

Update and Enhance: Implement updates based on user feedback.

Bug Fixes and Patches: Regularly fix any bugs or security issues.

5 Technologies to be used

5.1 Software Platform

a) Front-end

- HTML5: For structuring the content on the web pages.

- CSS3: For styling and making the web pages visually appealing and responsive.

- JavaScript: For adding interactivity and dynamic elements to the web pages.

-Bootstrap: For responsive design and pre-built components to ensure cross-device compatibility.

b) Back-end

- Python: The primary programming language used for server-side logic.

- Flask: A lightweight web framework for building the web application's backend.

- PyPDF2: For extracting text from PDF files.

- gTTS (Google Text-to-Speech): For converting extracted text to speech (audio files).

- qrcode: For generating QR codes from the provided content.

- Jinja2: The templating engine used by Flask for rendering HTML templates.

5.2 Hardware Platform

Processor	Intel I5 11th gen.
Hard Disk Space	512 GB
Ram Memory	8 GB
Operating System	Windows 11
Editor	VS Code

5.3 Tools, if any

Version Control:

Git: For version control and managing the source code.

GitHub: For hosting the code repository and collaboration.

Design Tools:

Figma: For designing UI/UX wireframes and mockups.

Adobe XD: For creating detailed UI/UX designs.

Development Tools:

Postman: For testing API endpoints and backend logic.

Docker: For containerizing the application to ensure consistent environments across development and production.

Deployment Platforms:

Pythonanywhere: For deploying and hosting the application.

Render: An alternative deployment platform for web applications.

Vercel: Specifically useful for frontend hosting and serverless functions.

GitHub Pages: For hosting static parts of the site if needed.

6 Advantages of this Project

- Enhanced Accessibility: Makes written content accessible to visually impaired individuals.

- Time-Saving: Allows users to listen to content on the go, saving time and enabling multitasking.

- Auditory Learning: Supports auditory learners, improving comprehension and retention.

- Quick QR Code Generation: Simplifies sharing links, videos, audio files, and photos.

- User-Friendly Interface: Easy navigation and usage for non-technical users.

- Cost-Effective: Utilizes free hosting platforms, reducing costs.

- Multi-Functional: Combines PDF to audiobook conversion and QR code generation in one application.

- Customization: Generates QR codes for various types of content.

- Scalable Infrastructure: Handles increasing user demands without performance issues.

- Future Enhancements: Designed for expandability with potential for new features.

- Educational Value: Serves as a learning tool for web development skills.

- Easy Sharing: Facilitates quick information sharing through QR codes.

7 Assumptions, if any

- PDFs are primarily text-based and do not contain complex formatting or images.
- Users have a stable internet connection to access and use the application.
- Users are familiar with basic web application navigation and functionality.
- The application will be hosted on reliable and secure free hosting platforms.
- Users have compatible devices and browsers to access the application.
- The application will handle typical PDF file sizes and not extremely large documents.
- Users understand the purpose and use of QR codes for various types of content.
- The application will be continuously monitored and maintained for performance and security.

8 Future Scope and further enhancement of the Project

- OCR Implementation: Enable text extraction from image-based PDFs.
- Multi-Language Support: Expand text-to-speech conversion to multiple languages.
- Enhanced QR Customization: Offer more options for QR code customization.
- User Authentication: Implement user accounts for personalized experiences.
- Offline Mode: Enable access to converted files without internet.
- Cloud Storage Integration: Seamlessly access files from cloud storage services.
- Advanced Analytics: Gather insights for informed decision-making.
- Collaborative Features: Facilitate real-time collaboration on documents.
- Mobile Application: Develop companion apps for on-the-go access.
- Voice Commands: Enable hands-free interaction with the app.
- Gamification: Introduce gamified elements for user engagement.
- Community Contributions: Foster collaboration and feedback from users.

9 Project Repository Location

S#	**Project Artifacts (softcopy)**	**Location** (Folder Name, Drive Link etc.)	**Verified by Project Guide**	**Verified by HOD**
1.	Project Synopsis Report (Final Version)	https://github.com/itsharshitsuman/AudiCode/blob/main/Project%20Report%20AudiCode.docx	Name and Signature	Name and Signature
2.	Project Progress updates		Name and Signature	Name and Signature
3.	Project Report (Final Version)	https://github.com/itsharshitsuman/AudiCode/blob/main/Project%20Report%20AudiCode.docx	Name and Signature	Name and Signature
4.	Git Repository	https://github.com/itsharshitsuman/AudiCode.git	Name and Signature	Name and Signature
5.	certificate Copy	https://drive.google.com/file/d/1_-mQlaM7zbcjzLHpbwZuU7GTLBClsG8a/view?usp=drive_link	Name and Signature	Name and Signature

10 Definitions, Acronyms, and Abbreviations

Abbreviation	**Description**
PDF	Portable Document Format
OCR	Optical Character Recognition
ML	Machine Learning
UI	User Interface
UX	User Experience
IoT	Internet of Things
API	Application Programming Interface
URL	Uniform Resource Locator
TTS	Text-to-Speech
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
QR Code	Quick Response Code
Flask	Web application framework for Python
HTTPS	Hypertext Transfer Protocol Secure
PyPDF2	Python library for PDF manipulation
gTTS	Google Text-to-Speech
GitHub	Online platform for code hosting and collaboration
CLI	Command Line Interface

11 Conclusion

AudiCode represents a significant advancement in digital content accessibility and efficiency by providing a comprehensive solution for converting PDFs to audiobooks and generating QR codes. Through the integration of innovative technologies and user-centric design, AudiCode offers a user-friendly platform that caters to diverse user needs, including those of visually impaired individuals, auditory learners, and users seeking quick information sharing.

By leveraging text-to-speech conversion and QR code generation capabilities, AudiCode enhances accessibility, convenience, and learning flexibility for users, while also promoting efficient information sharing and collaboration. The project's future scope and potential for further enhancement underscore its commitment to continuous improvement and adaptation to evolving user needs and technological advancements.

In conclusion, AudiCode exemplifies the power of technology to transform digital content consumption and sharing, making it more inclusive, efficient, and accessible to all users. As AudiCode continues to evolve and grow, it will undoubtedly play a significant role in facilitating digital communication, learning, and collaboration in various contexts.

12 References

1. Flask Documentation: https://flask.palletsprojects.com/

2. PyPDF2 Documentation: https://pythonhosted.org/PyPDF2/

3. gTTS Documentation: https://gtts.readthedocs.io/

4. qrcode Documentation: https://pypi.org/project/qrcode/

5. Bootstrap Documentation: https://getbootstrap.com/docs/5.0/getting-started/introduction/

6. Figma: https://www.figma.com/

7. Adobe XD: https://www.adobe.com/products/xd.html

8. GitHub Pages: https://pages.github.com/

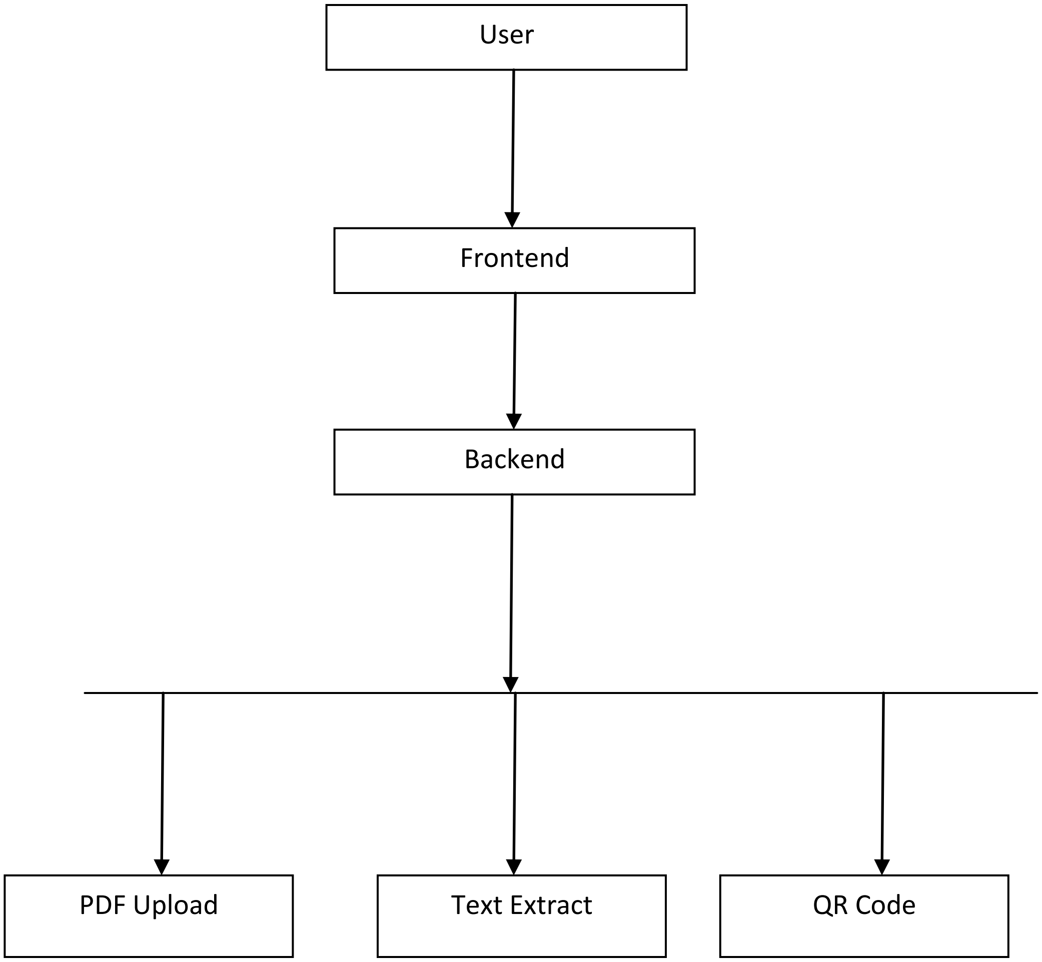
9. Pythonanywhere: https://www.pythonanywhere.com/

10. Render: https://render.com/

11. Vercel: https://vercel.com/

These references were used for documentation, libraries, frameworks, and hosting platforms to develop and deploy the AudiCode project.

Annexure A
Data Flow Diagram (DFD)



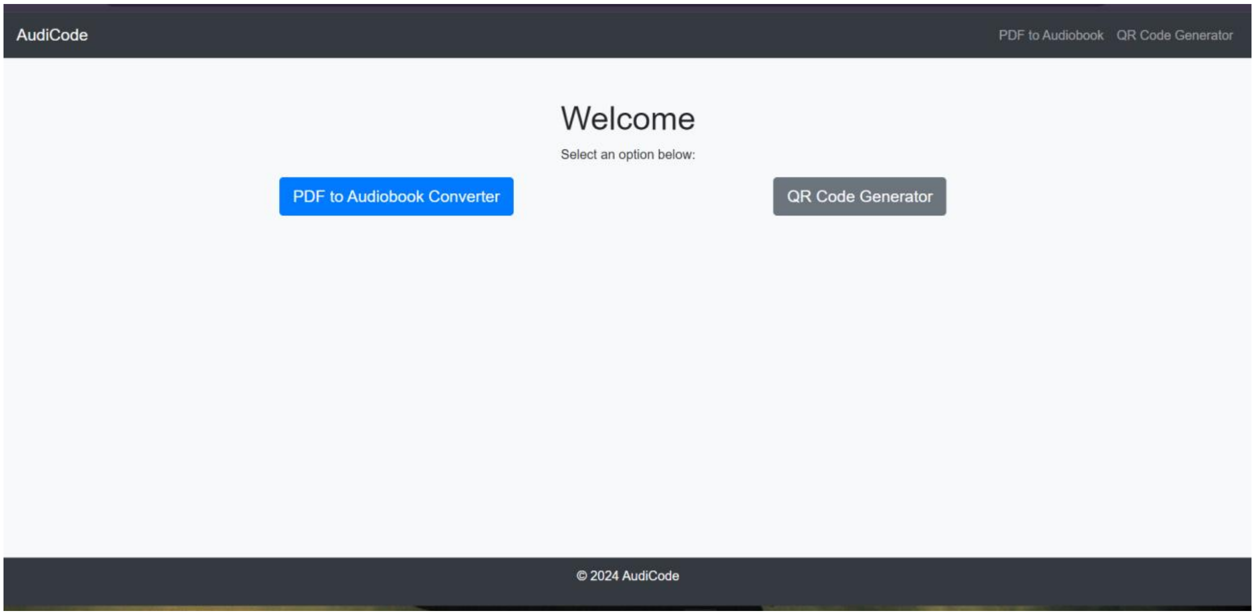
Annexure B
Data Dictionary (DD)

Example:
User Table (USR)

Field Name	Description	Data Type
PDF File	Uploaded PDF file for conversion to audiobook.	File
Text Content	Extracted text from the PDF file.	String
Audio File	Generated audio file (audiobook) from the extracted text.	File
QR Code Content	Content (e.g., link, text) to be encoded into a QR code.	String
QR Code Image	Generated QR code image based on the QR code content.	Image File

Annexure C
Screen Shots

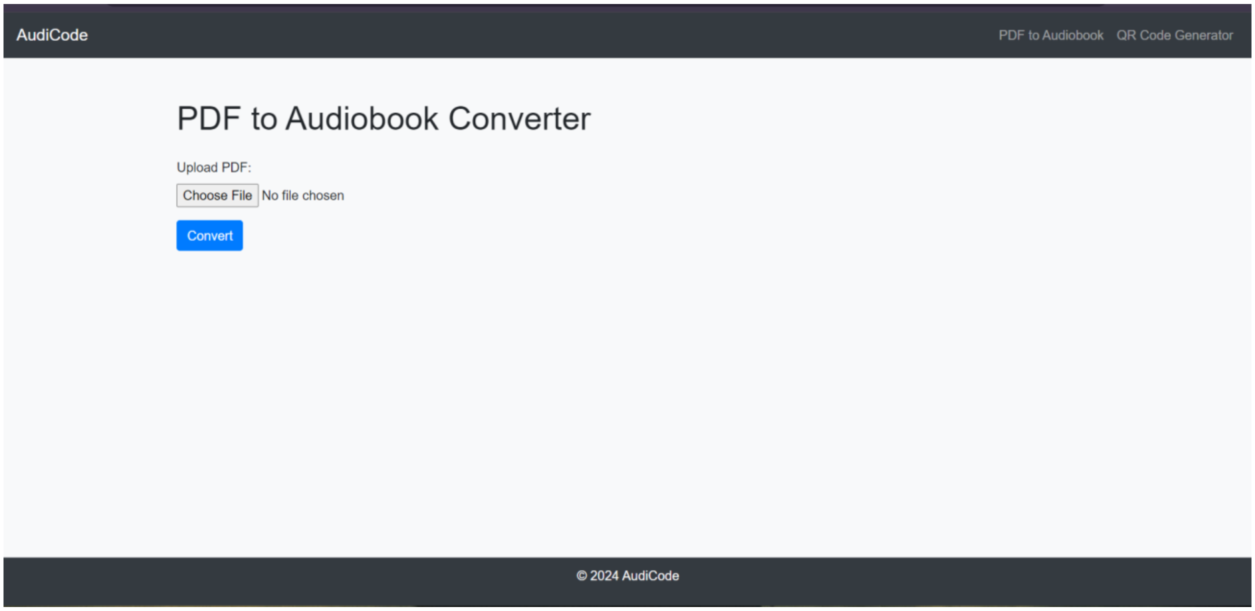
Home Page:



Welcome

Select an option below:

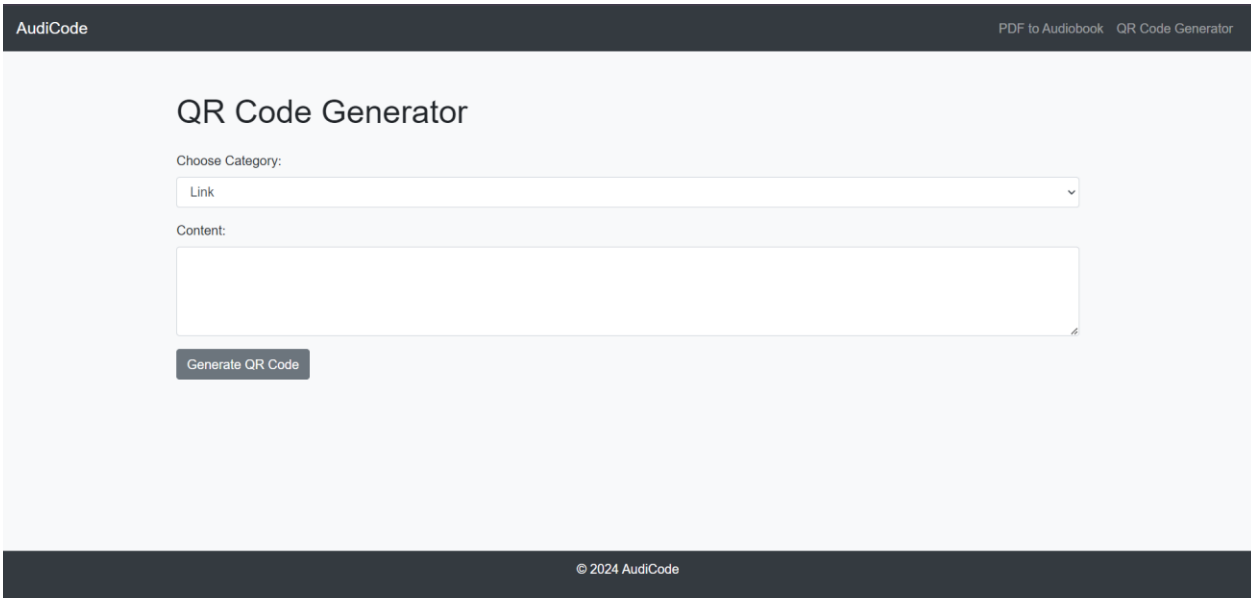
PDF to Audiobook converter page:



PDF to Audiobook Converter

Upload PDF:

QR Code Generator page:



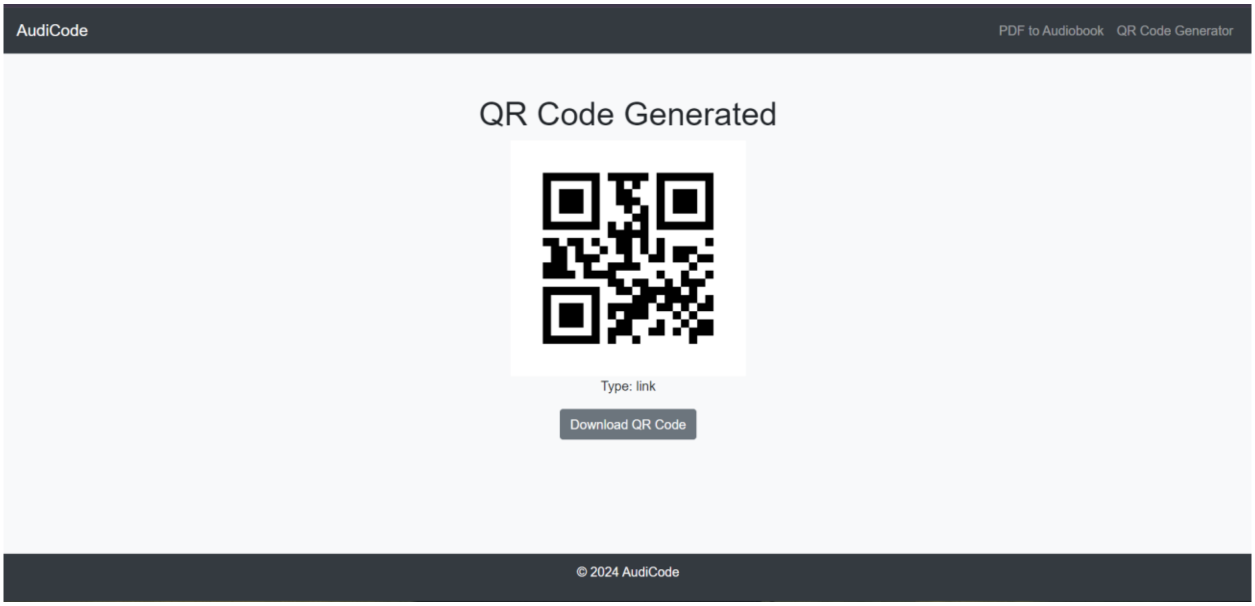
QR Code Generator

Choose Category:

Link

Content:

QR Code Download page:



QR Code Generated

Type: link

Annexure D

Source Code

App.py-

```
from flask import Flask, request, jsonify, send_from_directory, render_template, send_file, redirect, url_for
import os
from PyPDF2 import PdfReader
from gtts import gTTS
import qrcode
import logging
app = Flask(__name__)
UPLOAD_FOLDER = 'uploads'
AUDIO_FOLDER = 'audio'
QR_FOLDER = 'static'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
os.makedirs(AUDIO_FOLDER, exist_ok=True)
os.makedirs(QR_FOLDER, exist_ok=True)
logging.basicConfig(level=logging.DEBUG)
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/pdf_to_audio')
def pdf_to_audio():
    return render_template('pdf_to_audio.html')
@app.route('/upload', methods=['POST'])
def upload_file():
    if 'pdf' not in request.files:
        return jsonify({'error': 'No file part'}), 400
    file = request.files['pdf']
    if file.filename == '':
        return jsonify({'error': 'No selected file'}), 400
    if file and file.filename.endswith('.pdf'):
        pdf_path = os.path.join(UPLOAD_FOLDER, file.filename)
        file.save(pdf_path)
       try:
            text = extract_text_from_pdf(pdf_path)
            if not text:
                raise ValueError("Extracted text is empty")
            audio_filename = file.filename.replace('.pdf', '.mp3')
            audio_path = convert_text_to_audio(text, audio_filename)
            return redirect(url_for('download_audio', filename=audio_filename))
        except Exception as e:
            logging.error(f"Error processing file {file.filename}: {str(e)}")
            return jsonify({'error': f"Failed to process the PDF: {str(e)}"}), 500
    return jsonify({'error': 'Invalid file format'}), 400
def extract_text_from_pdf(pdf_path):
    logging.debug(f"Extracting text from PDF: {pdf_path}")
  text = ""
    try:
        with open(pdf_path, 'rb') as pdf_file:
            pdf_reader = PdfReader(pdf_file)
            for page_num, page in enumerate(pdf_reader.pages):
                page_text = page.extract_text()
                logging.debug(f"Extracted text from page {page_num}: {page_text[:100]}...")  # Log first 100 characters
                text += page_text
    except Exception as e:
        logging.error(f"Failed to extract text from PDF {pdf_path}: {str(e)}")
        raise
    return text
def convert_text_to_audio(text, filename):
    logging.debug(f"Converting text to audio: {filename}")
    try:
        tts = gTTS(text=text, lang='en')
        audio_path = os.path.join(AUDIO_FOLDER, filename)
        tts.save(audio_path)
    except Exception as e:
        logging.error(f"Failed to convert text to audio {filename}: {str(e)}")
        raise
    return audio_path
@app.route('/audio/<filename>')
def download_audio(filename):
    return send_file(os.path.join(AUDIO_FOLDER, filename), as_attachment=True)


@app.route('/qr_generator')
def qr_generator():
    return render_template('qr_generator.html')
@app.route('/generate_qr', methods=['POST'])
def generate_qr():
    content_type = request.form['content_type']
    content = request.form['content']
    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size=10,
        border=4,
    )
    qr.add_data(content)
    qr.make(fit=True)

    img = qr.make_image(fill_color="black", back_color="white")
    img.save(os.path.join(QR_FOLDER, 'qrcode.png'))

    return render_template('result.html', content_type=content_type,
download_link='/download_qr')

@app.route('/download_qr')
def download_qr():
    return send_file(os.path.join(QR_FOLDER, 'qrcode.png'), as_attachment=True)
if __name__ == '__main__':
    app.run(debug=True)
```

index.html-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f8f9fa;
        }
        .container {
            margin-top: 50px;
        }
        .navbar {
            background-color: #343a40;
        }
        .navbar-brand, .navbar-nav .nav-link {
            color: #ffffff;
        }
        .footer {
            background-color: #343a40;
            color: #ffffff;
            text-align: center;
            padding: 10px;
            position: absolute;
            bottom: 0;
            width: 100%;
        }
        .btn-primary {
            background-color: #007bff;
            border-color: #007bff;
        }
        .btn-primary:hover {
            background-color: #0056b3;
            border-color: #0056b3;
        }
```

```
        .btn-secondary {
            background-color: #6c757d;
            border-color: #6c757d;
        }
        .btn-secondary:hover {
            background-color: #545b62;
            border-color: #545b62;
        }
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark">
        <a class="navbar-brand" href="/">AudiCode</a>
        <div class="collapse navbar-collapse">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item"><a class="nav-link" href="/pdf_to_audio">PDF to Audiobook</a></li>
                <li class="nav-item"><a class="nav-link" href="/qr_generator">QR Code Generator</a></li>
            </ul>
        </div>
    </nav>
    <div class="container text-center">
        <h1>Welcome</h1>
        <p>Select an option below:</p>
        <div class="row">
            <div class="col-md-6">
                <a href="/pdf_to_audio" class="btn btn-primary btn-lg">PDF to Audiobook Converter</a>
            </div>
            <div class="col-md-6">
                <a href="/qr_generator" class="btn btn-secondary btn-lg">QR Code Generator</a>
            </div>
        </div>
    </div>
    <footer class="footer">
        <p>&copy; 2024 AudiCode</p>
    </footer>
</body>
</html>
```

pdf_to_audio.html-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PDF to Audiobook Converter</title>
    <link                href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
       <script>
        function showSpinner() {
            document.getElementById('spinner').style.display = 'block';
            document.getElementById('submit-button').disabled = true;
        }
    </script>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark">
```

```
        <a class="navbar-brand" href="/">AudiCode</a>
        <div class="collapse navbar-collapse">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item"><a class="nav-link" href="/pdf_to_audio">PDF to Audiobook</a></li>
                <li class="nav-item"><a class="nav-link" href="/qr_generator">QR Code Generator</a></li>
            </ul>
        </div>
    </nav>
    <div class="container">
        <h1>PDF to Audiobook Converter</h1>
        <form action="/upload" method="post" enctype="multipart/form-data" class="mt-4" onsubmit="showSpinner();">
            <div class="form-group">
                <label for="pdf">Upload PDF:</label>
                <input type="file" name="pdf" id="pdf" accept=".pdf" class="form-control-file" required>
            </div>
            <button type="submit" class="btn btn-primary" id="submit-button">Convert</button>
            <div class="spinner-border text-primary mt-3" id="spinner" role="status">
                <span class="sr-only">Loading...</span>
            </div>
        </form>
    </div>
    <footer class="footer">
        <p>&copy; 2024 AudiCode</p>
    </footer>
</body>
</html>
```

qr_generator.html-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>QR Code Generator</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
 </head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark">
    <a class="navbar-brand" href="/">AudiCode</a>
    <div class="collapse navbar-collapse">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item"><a class="nav-link" href="/pdf_to_audio">PDF to Audiobook</a></li>
        <li class="nav-item"><a class="nav-link" href="/qr_generator">QR Code Generator</a></li>
      </ul>
    </div>
  </nav>
  <div class="container">
    <h1>QR Code Generator</h1>
    <form action="/generate_qr" method="post" class="mt-4">
      <div class="form-group">
        <label for="content_type">Choose Category:</label>
        <select name="content_type" id="content_type" class="form-control">
          <option value="link">Link</option>
          <option value="video">Video</option>
          <option value="audio">Audio</option>
          <option value="photo">Photo</option>
        </select>
      </div>
      <div class="form-group">
        <label for="content">Content:</label>
        <textarea name="content" id="content" rows="4" class="form-control" required></textarea>
      </div>
      <button type="submit" class="btn btn-secondary">Generate QR Code</button>
    </form>
  </div>
  <footer class="footer">
    <p>&copy; 2024 AudiCode</p>
  </footer>
</body>
</html>
```

result.html-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Result</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
 </head>
```

```
<body>
  <nav class="navbar navbar-expand-lg navbar-dark">
    <a class="navbar-brand" href="/">AudiCode</a>
    <div class="collapse navbar-collapse">
      <ul class="navbar-nav ml-auto">
        <li class="nav-item"><a class="nav-link" href="/pdf_to_audio">PDF to Audiobook</a></li>
        <li class="nav-item"><a class="nav-link" href="/qr_generator">QR Code Generator</a></li>
      </ul>
    </div>
  </nav>
  <div class="container text-center">
    <h1>QR Code Generated</h1>
    <img src="/static/qrcode.png" alt="QR Code">
    <p>Type: {{ content_type }}</p>
    <a href="{{ download_link }}" class="btn btn-secondary">Download QR Code</a>
  </div>
  <footer class="footer">
    <p>&copy; 2024 AudiCode</p>
  </footer>
</body>
</html>
```