ALL-IN-ONER

MINOR PROJECT FILE

Project Details

Project Name : ALL-IN-ONER

Project Description : The primary goal of ALL-IN-ONER is to provide a **comprehensive suite of tools** that make common digital tasks—such as downloading videos, compressing images, merging PDFs, and generating QR or barcodes—easily accessible in one place. Each tool within the application is designed to handle specific user needs without requiring any additional software installation. Users can access the platform via any web browser, perform the tasks they need, and download their results directly.

Group members :

- Devender (10622069)
- Dhruv Gupta (10622072)
- Dhruv Kumar (10622073)
- Harsh (10622093)

Faculty Name : Mr. J.P SOJA

MINOR PROJECT

ALL-IN-ONER

Introduction

In the digital age, users often find themselves performing a variety of tasks that require specialized tools, such as downloading media, converting files, or generating scannable codes. However, switching between different tools for each task can be inefficient and time-consuming. **ALL-IN-ONER** is a multi-functional web-based platform designed to address this problem by offering a suite of tools within a single application. It brings together multiple functionalities, including video and audio downloading, barcode and QR code generation, PDF merging, image compression, and more. By consolidating these diverse tools into one platform, ALL-IN-ONER makes digital workflows simpler, faster, and more user-friendly.

Developed using the Flask web framework and powered by Python libraries, this project demonstrates the capability of web applications to offer a wide range of utilities that cater to various media, file management, and content creation needs. Whether the user is a student, a professional, or someone managing multimedia content, ALL-IN-ONER is designed to be accessible and functional for a broad audience. This project also demonstrates an efficient approach to file management by eliminating the need for multiple installations of software or the use of complex command-line tools.

Description:

The **ALL-IN-ONER** project is a powerful, user-friendly web platform built using Flask and Python. It provides a wide range of essential digital utilities designed to make everyday tasks more convenient. From media downloading to file management, this project integrates several useful features into a single platform, allowing users to access everything in one place.

Key Features:

1. **YouTube Video and Audio Downloader (MP4/MP3)**:
    - Download YouTube videos in MP4 format with ease.
    - Extract and download audio as MP3 from YouTube videos.
    - Real-time progress tracking while downloading.
    - Supports high-quality video/audio downloads.

2. **QR Code Generator**:

- Generate QR codes for any input data (URLs, text, etc.).
- Download the generated QR code as a PNG image for use in documents, websites, or presentations.

3. **Barcode Generator**:

- Generate barcodes in various formats like EAN-13, UPC, etc.
- Validate input codes and generate high-quality barcode images for product labels or other uses.
- Download the barcode image as a PNG file.

4. **PDF Merger**:

- Upload and merge multiple PDF files into a single PDF document.
- Simple drag-and-drop functionality for merging documents seamlessly.

5. **Image Compressor**:

- Compress large images to reduce file size without compromising quality.
- Download the compressed image, making it easier to share or upload.

6. **Instagram Reel Downloader**:

- Download Instagram reels by simply pasting the reel URL.
- No login or password required.

    - Save high-quality reels to your local device.

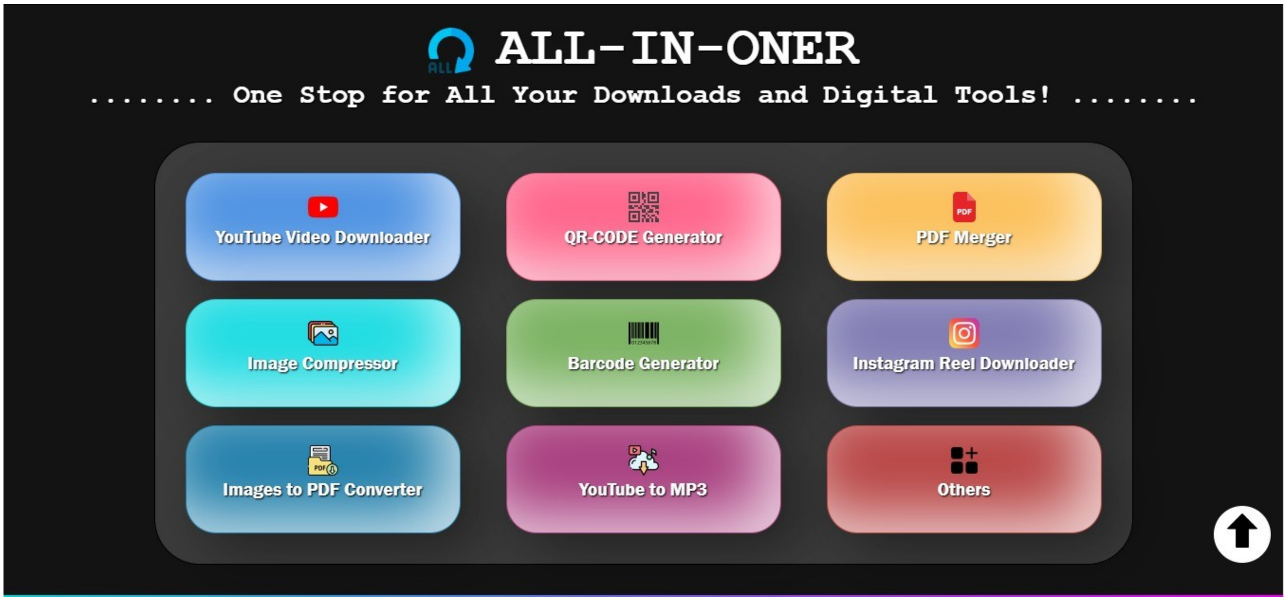
7. **Image to PDF Converter**:

    - Convert multiple images into a single PDF document.
    - Supports various image formats (JPEG, PNG, etc.).
    - Useful for creating image-based reports or digital documents.

8. **YouTube to MP3 Downloader**:

- Quickly convert and download YouTube videos as MP3 audio files.
- Ideal for users who want to listen to music or audio content offline without video.
- Easy-to-use interface for seamless conversion.

HOME PAGE



ALL-IN-ONER

.... One Stop for All Your Downloads and Digital Tools! ....

The home page of the ALL-IN-ONER website presents a sleek and user-friendly interface, providing a comprehensive suite of digital tools designed to simplify everyday tasks. The layout is structured in a grid format with different colored boxes, each representing a distinct tool. The page title, ALL-IN-ONER, is displayed prominently at the top, with the tagline "One Stop for All Your Downloads and Digital Tools!" This clearly reflects the purpose of the platform: to offer multiple utilities in one place.

Key Sections:

1. **YouTube Video Downloader**: Represented with a blue button and YouTube's logo, this tool allows

users to download videos directly from YouTube in a convenient and straightforward way.

2. **QR Code Generator**: This pink section offers a tool for generating QR codes for links, text, or any other data, making it easy to share information via scannable codes.

3. **PDF Merger**: The yellow box indicates a tool to combine multiple PDF files into one, streamlining the document organization process.

4. **Image Compressor**: Located in a cyan box, this tool is designed to compress images, reducing file size without compromising quality.

5. **Barcode Generator**: Represented by a green button, this allows users to create barcodes in various formats for different applications.

6. **Instagram Reel Downloader**: The purple box links to a tool that enables users to download Instagram reels using a simple URL.

7. **Images to PDF Converter**: Another tool, indicated by a blue box, that converts multiple images into a single PDF document.

8. **YouTube to MP3**: This pink box is dedicated to converting YouTube videos to MP3 format, allowing users to download audio files easily.

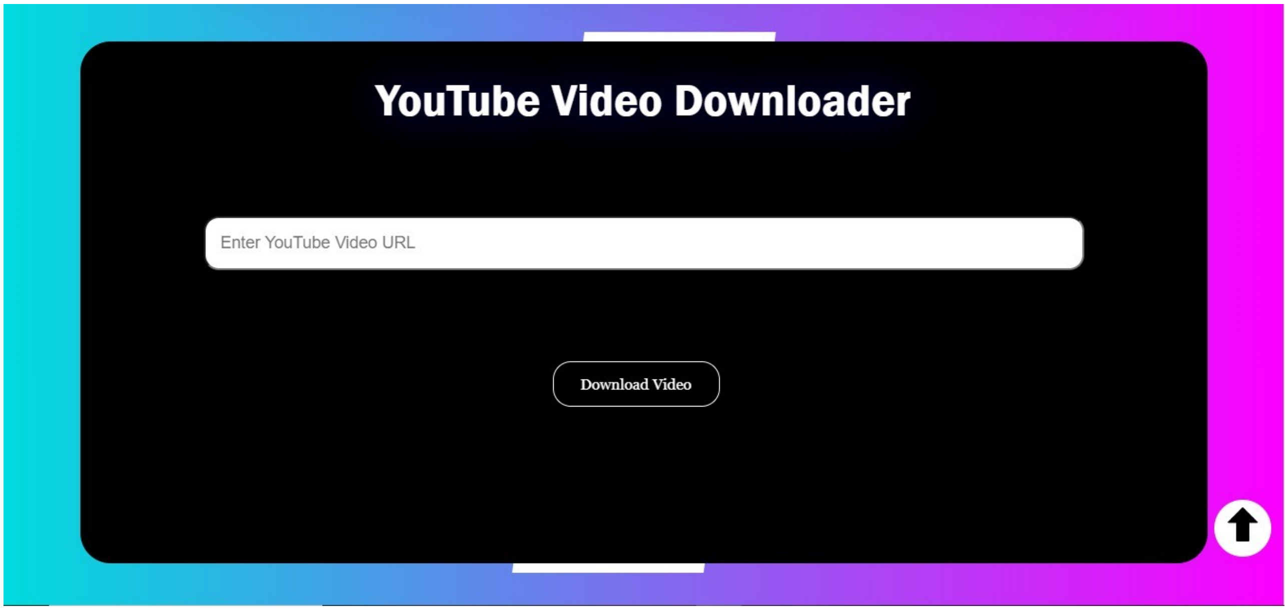
9. **Others**: The red box at the bottom is reserved for additional tools, hinting at more functionalities being provided on the website.

Design Elements:

- The overall theme uses dark background tones with colorful buttons that stand out, ensuring ease of navigation.
- Icons representing each tool enhance the visual appeal and provide an intuitive user experience.
- The scroll-up icon at the bottom of the screen indicates smooth navigation and user control, making it easier to return to the top of the page.

This home page offers a polished and well-organized interface that emphasizes functionality while maintaining a clean, attractive aesthetic. Each tool is readily accessible, allowing users to perform various tasks efficiently.

YouTube Video Downloader



YouTube Video Downloader

Enter YouTube Video URL

The YouTube Video Downloader page from the ALL-IN-ONER website provides a clean and minimalistic interface that enables users to download YouTube videos effortlessly. The page's design features a sleek black background framed by a gradient of cyan and magenta, creating a visually appealing contrast.

Key Elements:

1. **Title**: At the top of the page, the title **"YouTube Video Downloader"** is prominently displayed in bold, white text, making it clear what functionality the page offers.

2. **Input Field**: Just below the title, there's a single input field where users can enter the YouTube video URL they wish to download. The placeholder text reads **"Enter YouTube Video URL"**, ensuring users know exactly what to input.

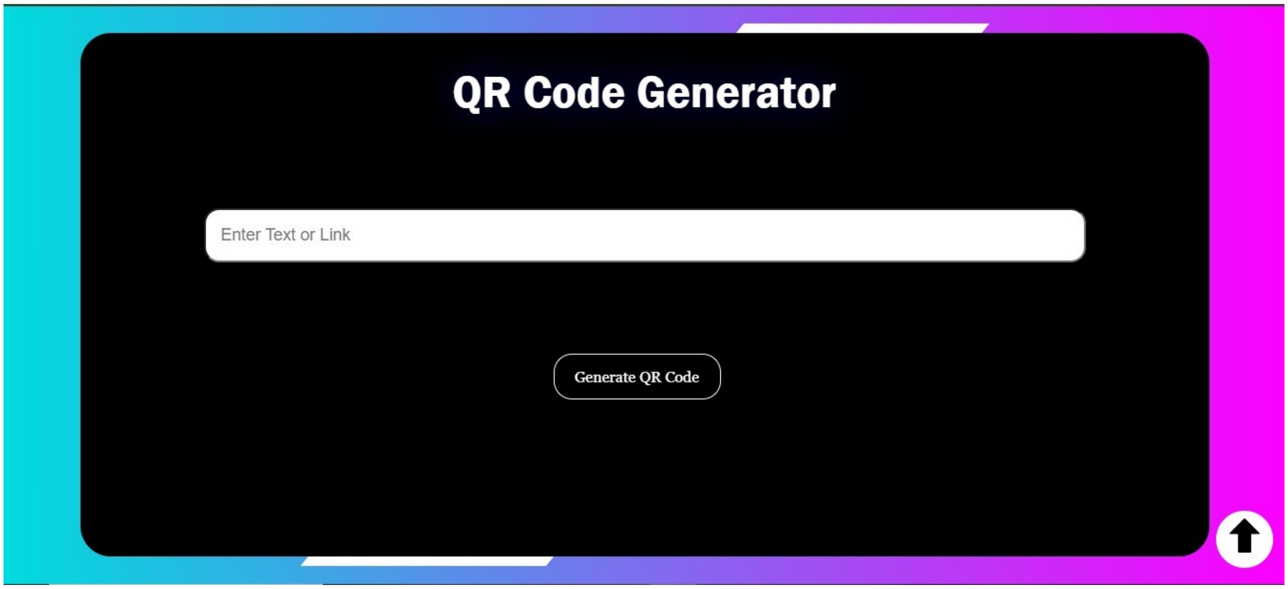
3. **Download Button**: Beneath the input field is a black button outlined in white, labeled **"Download Video"**. Clicking this button initiates the download process for the video, based on the URL provided.

Design Features:

- **Simplicity**: The overall layout is kept simple and intuitive, allowing users to focus on the primary task without distractions. The lack of unnecessary elements ensures smooth usability.
- **Visual Contrast**: The dark central area with a colorful gradient border makes the page modern and aesthetically pleasing, while the stark contrast highlights the key action areas (input field and button).
- **Ease of Use**: Users can easily navigate the page by pasting the YouTube URL and clicking the button to start the download. This simplicity makes the tool accessible for all users.

This page perfectly aligns with the minimalist and functional design philosophy of the **ALL-IN-ONER** platform, offering an effective solution for downloading YouTube videos with a user-friendly interface.

QR Code Generator



The QR Code Generator page of the ALL-IN-ONER website is designed to provide users with a quick and efficient way to generate QR codes for various purposes, such as links, messages, contact information, or other data. The interface is sleek and user-friendly, ensuring that even users with minimal technical expertise can easily create QR codes.

Key Elements:

1. **Title**: The page is titled **"QR Code Generator"**, displayed in large, bold text at the top, clearly indicating the function of the tool.

2. **Input Field**: Below the title, there is a central input field where users can enter the text, link, or other information they want to convert into a QR code. The placeholder within the input box might read something like **"Enter text or URL to generate QR code"**,

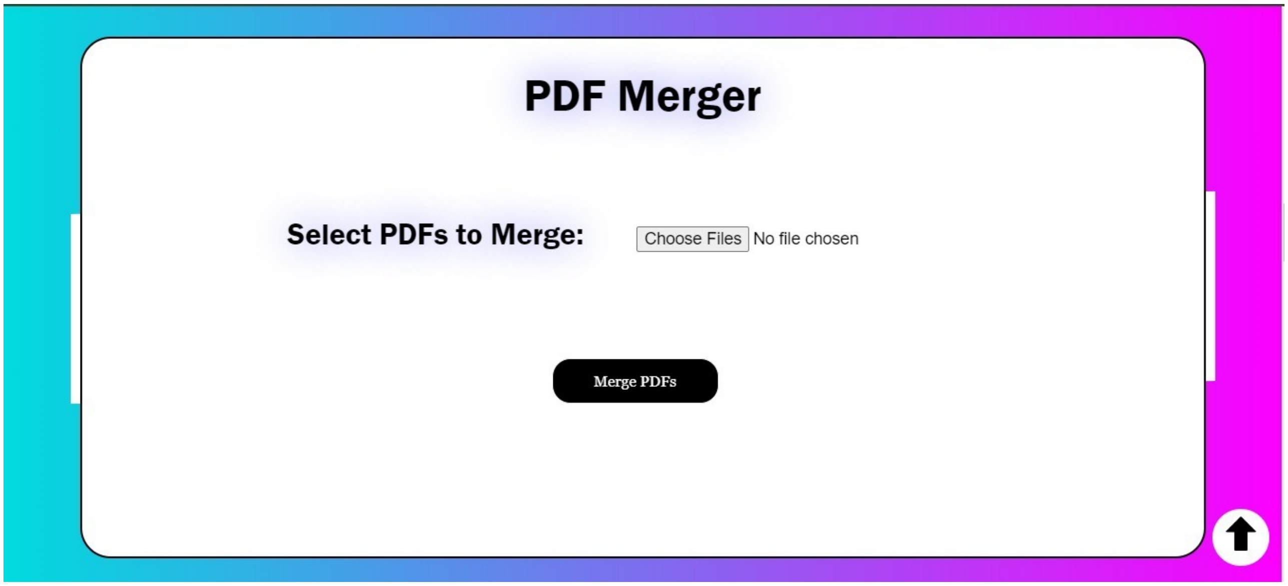
guiding users to input the correct information.

3. **Generate Button**: Once the user has entered the desired content, they can click the **"Generate QR Code"** button to instantly create a QR code based on the information provided.
4. **Download Option**: After generating the QR code, a downloadable version of the image is displayed, allowing users to save the QR code image to their device for future use.

Design Features:

- **Simplicity and Clarity**: The layout is simple and easy to understand. With only one field to fill and one button to click, users can generate QR codes with minimal effort.
- **Instant Feedback**: The QR code is generated instantly after clicking the button, and the design ensures the process is quick and smooth, with no unnecessary steps.
- **Versatility**: This tool is versatile enough to handle different types of data, making it a convenient solution for users who need QR codes for personal, professional, or promotional purposes.

PDF Merger



The PDF Merger page of the ALL-IN-ONER website is designed to help users easily merge multiple PDF files into a single document. This tool is particularly useful for individuals who frequently work with documents, allowing them to combine various PDF files such as reports, presentations, or study materials into one consolidated file.

Key Elements:

1. **Title**: The page prominently displays the title **"PDF Merger"**, ensuring users know they are on the right tool for merging PDF files.

2. **File Upload Area**: The core feature of the page is an intuitive **file upload area**, where users can drag and drop their PDF files or browse to upload them. The upload section may include a prompt like **"Upload PDFs to merge"**.

3. **Rearranging Files**: Once the files are uploaded, users can rearrange the order of the PDFs to suit their needs. The

interface is likely simple, with drag-and-drop functionality, enabling users to easily shift files into the desired order.

4. **Merge Button**: After arranging the files, the user can click the **"Merge PDF"** button to start the merging process. This action will quickly combine all the uploaded PDFs into one document.

5. **Download Merged File**: After the PDFs are merged, a download link is provided, allowing users to save the newly merged PDF file to their device for immediate use.

Design Features:

- **User-Focused Design**: The interface is kept simple to ensure ease of use. From uploading PDFs to downloading the final merged document, the process is straightforward and designed for efficiency.

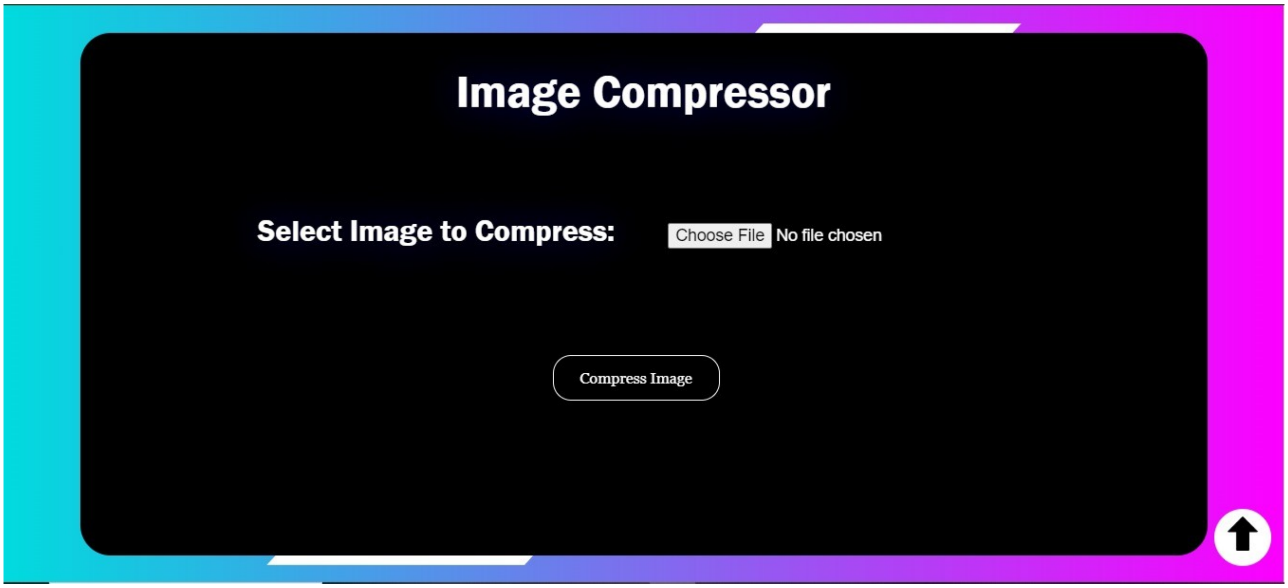
- **Support for Multiple Files**: Users can upload multiple PDF files at once, making this a practical tool for those who frequently need to compile multiple documents.

- **Seamless Merging Process**: The merging process is quick and requires minimal user input, ensuring that users can complete their tasks without complications.

Overall Experience:

The **PDF Merger** on the **ALL-IN-ONER** platform is designed to make combining PDF files as simple as possible. Its clean interface and easy-to-use functionality offer a seamless experience for users, helping them merge their files quickly and efficiently with just a few clicks. Whether for professional or personal use, this tool is a time-saver for anyone who deals with large amounts of PDF documents.

IMAGE COMPRESSOR



The Image Compressor page of the ALL-IN-ONER website provides users with a powerful tool to reduce the file size of images without compromising on quality. This feature is essential for individuals who need to save storage space, reduce load times on websites, or share images online where file size restrictions apply.

Key Elements:

1. **Title**: The page is labeled **"Image Compressor"**, clearly indicating its purpose of compressing images to reduce their file size.

2. **File Upload Section**:

    - The user is presented with an intuitive file upload interface where they can easily **select an image** from their device.
    - Users can **drag and drop** or browse through their files to upload images in formats like **JPEG, PNG**, and other common image types.

3. **Automatic Compression**:

    - Once the image is uploaded, the **compression process begins**

    automatically.
    - The image is compressed to a **low-level file size**, ensuring a significant reduction in the file size without requiring the user to adjust any settings.
    - No manual selection for compression levels is needed; the process is designed to be as **user-friendly** and **quick** as possible.
4. **Download Option**:

    - After the compression is complete, users can simply click the **Download** button to save the compressed image to their device.
    - This makes the process streamlined, reducing any additional steps or complications.

Design Features:

- **Minimalist Interface**: The design is kept **simple** and focused, with clear instructions for uploading and downloading, making it accessible for all types of users.

- **No Settings to Adjust**: To keep the process effortless, there are **no extra compression level choices**. The system automatically applies a standard low-level compression, balancing size reduction with acceptable image quality.

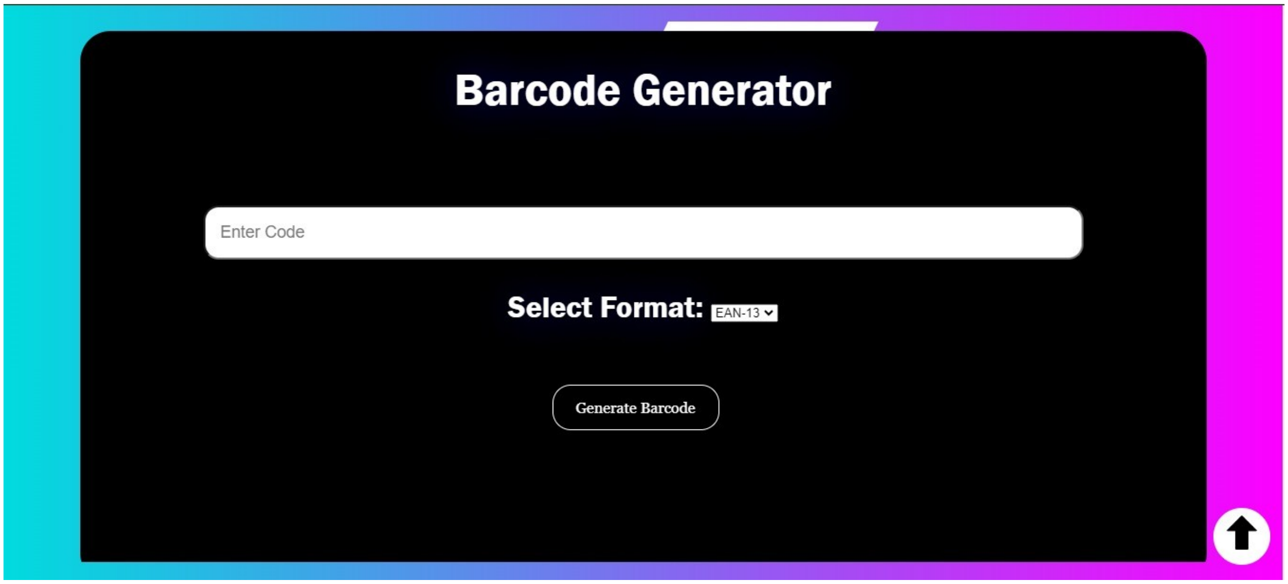
Benefits:

- **Fast and Simple**: The automatic process is designed for users who need quick results without any complexity. Just upload, and the image is compressed instantly.

- **Optimized File Size**: The image compression is designed to **drastically reduce file size**, making it easier to store, share, or upload the image online without worrying about large file sizes.

- **Perfect for Casual Use**: This tool is ideal for those who don't need detailed control over compression settings but just want a fast and effective way to reduce image size.

BARCODE GENERATOR



The **Barcode Generator** tool in the **ALL-IN-ONER** website allows users to easily generate barcodes by entering a numeric value and selecting a preferred barcode format. Here’s a detailed breakdown of the feature:

Key Elements:

1. **Title**: The page is labeled **"Barcode Generator"**, highlighting its core function, which is to create barcodes from numeric inputs.

2. **Number Input**:

   - Users are prompted to **enter a numeric value** that will be used to generate the barcode.
   - The input field allows for numbers compatible with the different barcode formats (EAN-13, UPC-A, EAN-8).

3. **Format Selection**:

   - Users can select from a **dropdown menu** or radio buttons that offer three different barcode formats:

        - **EAN-13**: A 13-digit barcode commonly used in retail.
        - **UPC-A**: A 12-digit barcode mainly used in the United States for consumer products.
        - **EAN-8**: A shorter 8-digit version of the EAN-13 format, used for smaller packaging.
    - Once the format is selected, the tool adapts to ensure the correct barcode type is generated.
4. **Generate Button**:

    - After entering the number and selecting the format, users click the **"Generate"** button.
    - The system processes the information and instantly creates the barcode in the selected format.
5. **Barcode Display and Download**:

    - Once the barcode is generated, it is **visually displayed** on the page for the user to see.
    - Users are then given the option to **download** the barcode in various formats like PNG or PDF, depending on their needs.

Design Features:

- **User-Friendly Interface**: The interface is clean and straightforward, with clear instructions for entering the number and selecting the barcode format.
- **No Complex Configurations**: The process is designed to be **hassle-free**, allowing even novice users to generate barcodes quickly.
- **Responsive Design**: The barcode generator is optimized for both desktop and mobile users, ensuring smooth performance across devices.

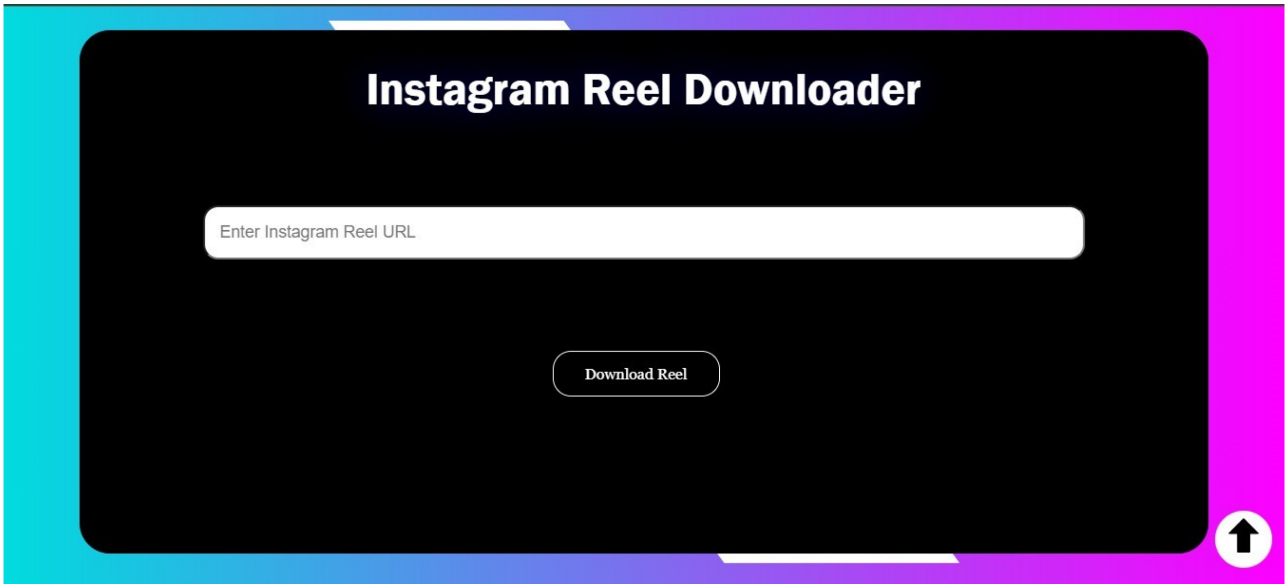
Benefits:

- **Customizable Barcode Generation**: Users have the flexibility to create barcodes in different standard formats based on their specific needs.
- **Instant Results**: The tool instantly generates the barcode, allowing users to view and download it without delay.
- **Easy Integration**: The downloaded barcodes can be seamlessly integrated into inventory systems, product labels, or any other business or personal use.

User Experience:

The **Barcode Generator** is designed to offer a **quick and efficient** solution for creating barcodes with minimal effort. Whether for retail, packaging, or personal projects, this tool provides users with the functionality they need to generate high-quality barcodes in a matter of seconds. With easy format selection and instant downloads, it offers a **smooth and user-friendly experience**.

Instagram Reel Downloader



The Instagram Reel Downloader tool in the ALL-IN-ONER website allows users to effortlessly download Instagram Reels directly by pasting the link. Here's an overview of how this tool works:

Key Features:

1. **Title**: The page is titled **"Instagram Reel Downloader"**, clearly stating its purpose—to download Instagram Reels easily.

2. **Input Field**:

    - Users are provided with a **text input box** where they can **paste the URL of the Instagram Reel** they want to download.
    - The input field is simple and straightforward, requiring only the link to the Reel for processing.

3. **Download Button**:

    - Once the link is entered, users simply click the

"Download Reel" button to initiate the download.
   - The system quickly processes the link and retrieves the video for the user.

4. **Video Format**:

   - The downloaded Instagram Reel is saved in its **original video quality**, ensuring that the user gets a high-resolution copy of the content.
   - The video is typically saved in **MP4 format**, which is compatible with a wide range of devices.

5. **No Additional Software Needed**:

   - There is no need for external apps or software. Everything is handled directly within the web page, making it convenient for users.

6. **Instant Download**:

   - The tool ensures **quick processing and downloading**, allowing users to get the Reels they want in seconds.

Design and User Experience:

- **Simple and Clean Layout**: The layout is minimalistic with a focus on ease of use. The interface is user-friendly, allowing anyone to download Instagram Reels without any technical knowledge.
- **Responsive**: Whether on mobile or desktop, the page adapts to the device for smooth performance.
- **Fast and Reliable**: The Reel download process is fast and reliable, ensuring users get their videos without delays or errors.

Benefits:

- **Easy Access to Instagram Content**: Users can download Instagram Reels to watch offline, share with others, or save for future reference.

- **High-Quality Downloads**: The downloader preserves the original quality of the video, ensuring that users receive clear, high-definition content.
- **No Sign-Up or Log-In Required**: There’s no need to create an account or sign in, making the process hassle-free and anonymous.

How It Works:

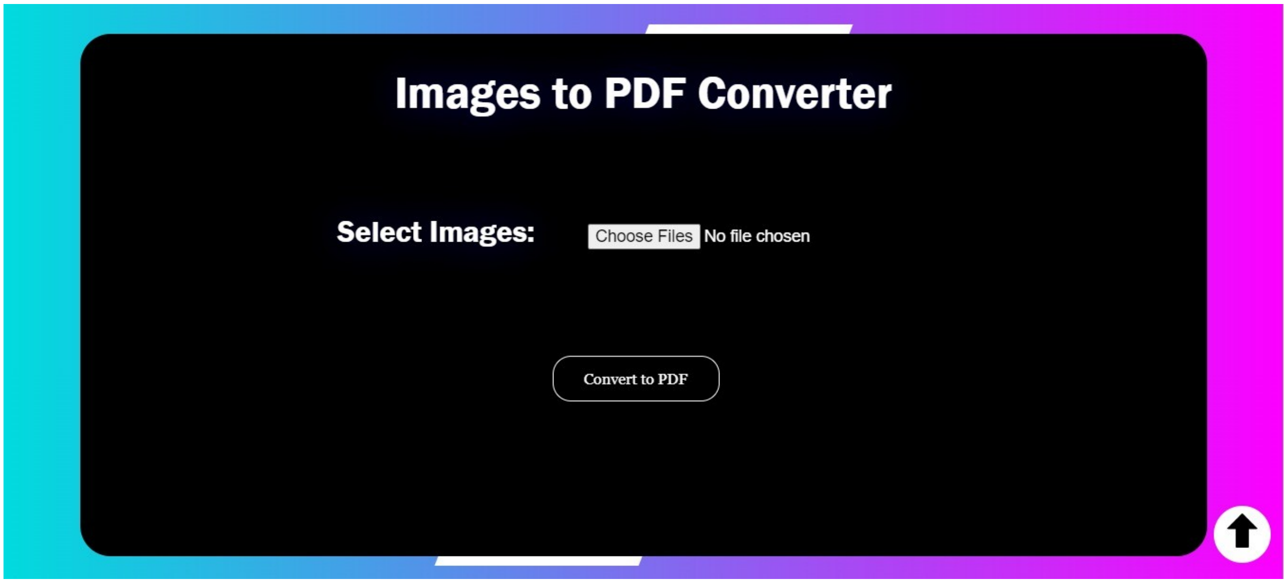
1. **Copy the Reel Link**: Users copy the link of the Instagram Reel they want to download.
2. **Paste and Download**: Paste the link into the input field, then hit the download button.
3. **Save the Reel**: The Reel is downloaded instantly and saved to the user’s device.

Use Cases:

- **Offline Viewing**: Save Instagram Reels to watch when you don’t have internet access.
- **Content Creation**: Use the Reels as inspiration or reference for your own projects.
- **Sharing**: Download and share Instagram Reels with friends or on other social platforms.

In summary, the **Instagram Reel Downloader** tool provides a simple and effective way to download and save Instagram Reels in high quality. It's a user-friendly solution for those who want to enjoy or repurpose Instagram content offline, with a focus on ease of use and fast performance.

Image to PDF Converter



Images to PDF Converter

Select Images: No file chosen

The Image to PDF Converter tool in the ALL-IN-ONER website allows users to easily convert their images into a PDF format. This is especially useful for creating professional documents, sharing multiple images in a single file, or preserving image quality in a more universal format. Below is an overview of how this tool works:

Key Features:

1. **Title**: The page is titled **"Images to PDF Converter"**, which clearly describes its functionality—converting image files to PDF format.

2. **Image Upload Section**:

   - Users are provided with a section where they can **upload multiple image files** (JPEG, PNG, etc.).
   - Users can **drag and drop** images or manually select files from their device.
3. **Processing and Conversion**:

  - Once the images are uploaded, the tool automatically processes the files and **converts them into a single PDF document**.
  - The images are maintained in their original quality, ensuring a high-resolution output in the PDF.

4. **Order and Arrangement**:

    - Users have the option to **arrange the order** of images before generating the PDF. This allows for better control over the flow of images in the final document.

5. **Download Button**:

    - After converting the images, the user can click the **“Download PDF”** button to download the generated PDF file to their device.
    - The downloaded PDF contains all the selected images in the specified order.

Design and User Experience:

- **Clean and Minimalist Interface**: The design is simple and intuitive, allowing even non-technical users to convert images to PDF effortlessly.
- **User-Friendly Uploads**: Uploading images is quick and smooth, with the tool supporting both drag-and-drop functionality and manual file selection.
- **Responsive Design**: The page is designed to work seamlessly on both desktop and mobile devices, making it accessible from any platform.
- **Fast and Efficient**: The conversion process is fast, ensuring minimal wait time for users.

Benefits:

- **Combining Multiple Images**: Users can combine multiple images into a single, cohesive PDF file, making it easier to share or present image content.

- **Universal PDF Format**: The output is in PDF, a universally accepted file format that ensures compatibility across devices and platforms.
- **No Quality Loss**: The converter maintains the original resolution and quality of the images, ensuring the best output in the final PDF.
- **No Account Required**: The tool does not require any sign-up or log-in, providing a seamless, no-hassle experience.

How It Works:

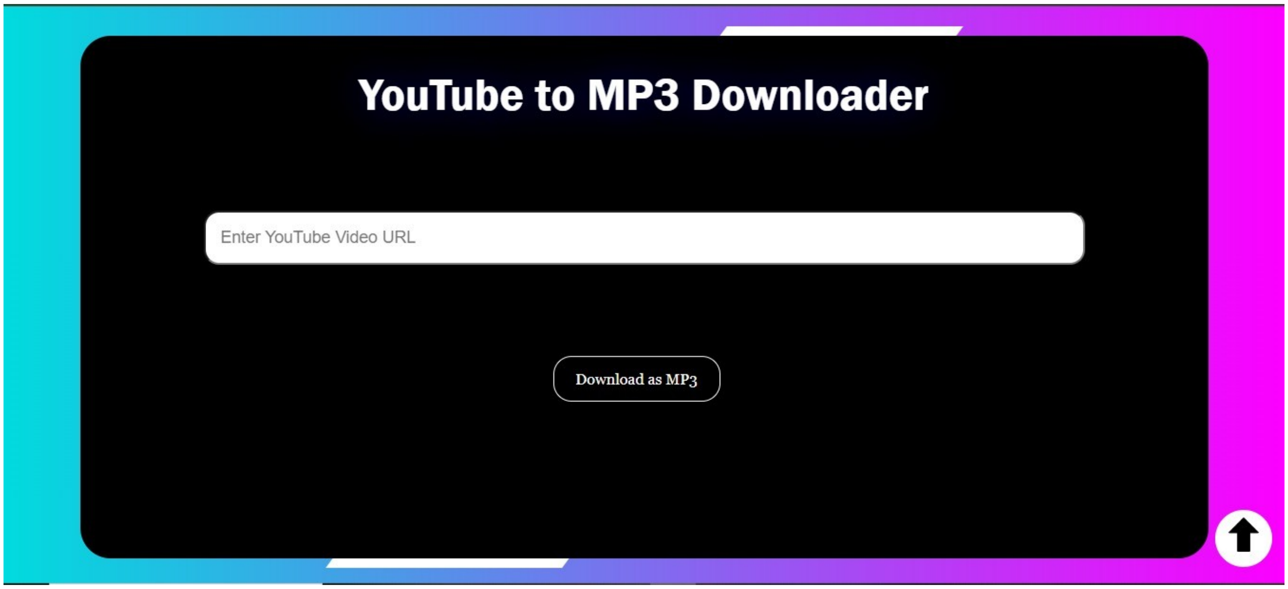
1. **Upload the Images**: Users can select or drag and drop multiple image files they wish to convert.
2. **Arrange the Order**: If necessary, users can rearrange the order of the images before generating the PDF.
3. **Download the PDF**: After the conversion is completed, the PDF file is ready for download.

Use Cases:

- **Document Creation**: Combine multiple images into a single PDF for reports, presentations, or sharing important files.
- **Photo Albums**: Create a digital photo album by converting multiple images into a well-organized PDF.
- **Sharing and Archiving**: Easily share or archive a collection of images in a single, universally accessible PDF file.

In summary, the **Image to PDF Converter** tool provides an easy and efficient way to combine multiple images into a single, professional PDF document. Its simple interface, fast processing, and high-quality output make it an essential tool for users who need to convert images to PDF without any hassle.

YOUTUBE TO MP3 DOWNLOADER



YouTube to MP3 Downloader

The YouTube to MP3 Downloader tool on the ALL-IN-ONER website enables users to easily extract audio from YouTube videos and download it in MP3 format. This is a popular feature for users who want to enjoy music, podcasts, or other audio content from YouTube without the need for video playback. Here's a detailed overview of this tool:

Key Features:

1. **Title**: The page is clearly labeled **"YouTube to MP3 Downloader"**, indicating its primary function to convert YouTube videos into MP3 audio files.

2. **Input Field**:

    - Users are provided with a **text input field** labeled **"Enter YouTube Video URL"** where they can paste the link to the YouTube video they wish to convert.
    - This field is designed to accept valid YouTube URLs, ensuring a straightforward user experience.

3. **Download Button**:

    - After entering the video URL, users can click the **“Download MP3”** button to initiate the conversion process.
    - The button is prominently displayed for easy access.

4. **Processing and Conversion**:

    - Once the URL is submitted, the tool processes the video and extracts the audio track, converting it to MP3 format.
    - The conversion is typically fast, allowing users to quickly obtain their audio files.

5. **Quality Selection**:

    - Depending on the tool's features, users may have the option to select the desired audio quality (e.g., low, medium, high) before downloading.

Design and User Experience:

- **User-Friendly Interface**: The layout is simple and intuitive, making it easy for users to navigate and utilize the tool without prior experience.
- **Responsive Design**: The page is optimized for both desktop and mobile devices, ensuring accessibility for all users.
- **Fast Processing**: The tool is designed to minimize waiting times, providing users with quick access to their audio files.

Benefits:

- **Audio Extraction**: Users can enjoy their favorite music or audio content from YouTube without the distraction of video playback.
- **Portable MP3 Format**: MP3 files are widely compatible with various devices, including smartphones, tablets, and music players, making it easy to listen on the go.

- **No Software Installation**: The tool operates online, meaning users do not need to install any additional software or plugins.
- **Free of Charge**: Users can utilize the tool at no cost, making it an economical solution for audio downloading.

How It Works:

1. **Paste the URL**: Users simply copy the URL of the YouTube video they wish to convert and paste it into the input field.
2. **Click Download**: After entering the URL, users click the **“Download MP3”** button to begin the conversion process.
3. **Save the MP3**: Once the audio is extracted and converted, users will receive a link to download the MP3 file to their device.

Use Cases:

- **Music Downloads**: Users can extract their favorite songs from YouTube for offline listening.
- **Podcast Accessibility**: Individuals can download podcasts or interviews from YouTube for easy access while commuting or exercising.
- **Study Material**: Students can convert educational videos to audio format for convenient review while on the go.

In summary, the **YouTube to MP3 Downloader** tool offers an efficient and user-friendly way to convert YouTube videos into MP3 audio files. With its straightforward interface, quick processing, and compatibility with various devices, it serves as a valuable resource for users seeking to enjoy audio content from YouTube conveniently.

Project Structure Overview

1. **`app.py`**:

   - This is the main backend script of your Flask application. It acts as the central hub for handling all the routes and logic for the various features offered by the platform.
   - The script includes functions to handle user requests, process data, and return appropriate responses. Each feature (like downloading videos, generating QR codes, etc.) will have its dedicated route and logic in this file.

2. **`templates/`**:

   - This directory contains all the HTML templates used for the front-end of the application.
   - Each feature of the platform will have its own HTML file, allowing for a clean separation of presentation logic. Flask uses Jinja2 templating engine, which means you can also include dynamic content rendering based on user inputs or backend data.
   - Common templates may include a home page, forms for inputting URLs, and pages displaying results.

3. **`static/`**:

   - This folder stores all static assets that enhance the look and functionality of your web application.
   - You will include CSS files for styling, JavaScript files for any client-side logic (like form validation or interactivity), and images that may be used within your

app.
- Organizing assets into this directory ensures that they are easily accessible and managed separately from your dynamic content.

4. **`requirements.txt`**:

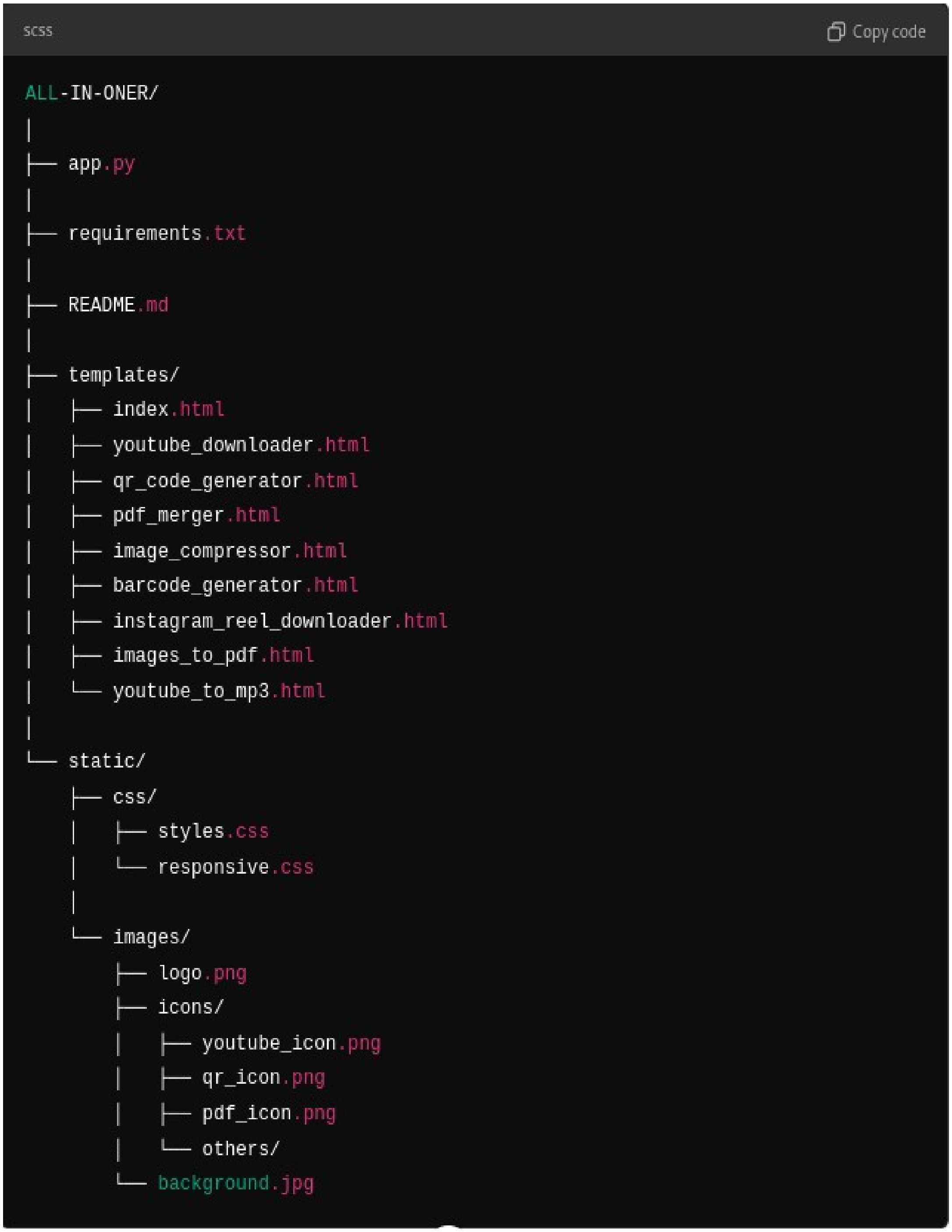
    - This file is crucial for specifying all Python dependencies required to run your project.
    - It includes libraries such as Flask (for web development), `yt-dlp` (for downloading YouTube videos), `qrcode` (for generating QR codes), and `Pillow` (for image processing).
    - Users can easily install all necessary packages by running `pip install -r requirements.txt` in their terminal.

5. **`README.md`**:

    - The README file provides a comprehensive overview of your project. It includes setup instructions, usage guides, and descriptions of each feature offered by the application.
    - It's essential for helping users understand how to install, configure, and use the platform. It may also contain information about any known issues, contributing guidelines, or acknowledgments.

Example File Structure

Here's how the directory structure might look :



```
ALL-IN-ONER/
│
├── app.py
│
├── requirements.txt
│
├── README.md
│
├── templates/
│   ├── index.html
│   ├── youtube_downloader.html
│   ├── qr_code_generator.html
│   ├── pdf_merger.html
│   ├── image_compressor.html
│   ├── barcode_generator.html
│   ├── instagram_reel_downloader.html
│   ├── images_to_pdf.html
│   └── youtube_to_mp3.html
│
└── static/
    ├── css/
    │   ├── styles.css
    │   └── responsive.css
    │
    └── images/
        ├── logo.png
        ├── icons/
        │   ├── youtube_icon.png
        │   ├── qr_icon.png
        │   ├── pdf_icon.png
        │   └── others/
        └── background.jpg
```

XSNDJNDCJLXDKL

INDEX.HTML

```html
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>All-in-One Flask Utility</title>

<!-- <style></style> -->


<!-- <link rel="stylesheet" href="style.css"> -->
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">

</head>
<body>
<section id="scroll"></section>
<div class="upperdiv">
<header>

<div class="headdiv">

<img id="headerimg" src= "{{ url_for('static', filename='images/all.png') }}" alt="">
<h1 id="h1head">ALL-IN-ONER</h1>

</div>

<p>........ One Stop for All Your Downloads and Digital Tools! ........</p>

<div class="arrow">


<a href="#scroll"><img id="arrowup" src="{{ url_for('static', filename='images/up-arrow.png') }}"
alt=""></a>


<!-- <img id="arrowup" src="{{ url_for('static', filename='images/up-arrow.png') }}" alt=""> -->
</div>


</header>
```

```
<div class="navclass">
<nav>


<div class="navbox" id="imgtag1" >

<img src="{{ url_for('static', filename='images/youtube.png') }}" class="imgclass">
<a class="link" href="#youtube-download">YouTube Video Downloader</a>
</div>



<div class="navbox" id="imgtag2" >

<img src="{{ url_for('static', filename='images/qr-code.png') }}" class="imgclass">
<a class="link" href="#qr-code-generator">QR-CODE Generator</a>
</div>


<div class="navbox" id="imgtag3" >

<img src= "{{ url_for('static', filename='images/pdf.png') }}" class="imgclass">
<a class="link" href="#pdf-merger">PDF Merger</a>
</div>


<div class="navbox" id="imgtag4" >


<img src= "{{ url_for('static', filename='images/gallery.png') }}" class="imgclass">
<a class="link" href="#image-compressor">Image Compressor</a>
</div>


<div class="navbox" id="imgtag5" >


<img src= "{{ url_for('static', filename='images/barcode-product.png') }}" class="imgclass">
<a class="link" href="#barcode-generator">Barcode Generator</a>
</div>


<div class="navbox" id="imgtag6" >


<img src= "{{ url_for('static', filename='images/instagram.png') }}" class="imgclass">
<a class="link" href="#instagram-downloader">Instagram Reel Downloader</a>
</div>
```

```html
<div class="navbox" id="imgtag7" >


<img src= "{{ url_for('static', filename='images/pdf-file.png') }}" class="imgclass">
<a class="link" href="#images-to-pdf">Images to PDF Converter</a>
</div>


<div class="navbox" id="imgtag8" >

<img src= "{{ url_for('static', filename='images/download.png') }}" class="imgclass">
<a class="link" href="#yt-mp3">YouTube to MP3</a>
</div>


<div class="navbox" id="imgtag9" >


<img src="{{ url_for('static', filename='images/menu.png') }}" class="imgclass">
<a class="link" href="#yt-mp3">Others</a>
</div>


</nav>




</div>

</div>


<div class="main">

<!-- YouTube Video Downloader -->
<section id="youtube-download" class="box">

<div class="in">

<h2>YouTube Video Downloader</h2>
<form action="/download" method="POST">
<!-- <label for="yt-url"></label> -->
<input type="text" id="yt-url" name="url" placeholder="Enter YouTube Video URL"
class="inputbar" required>
<button type="submit">Download Video</button>
</form>
```

```html
</div>

</section>

<!-- QR Code Generator -->
<section id="qr-code-generator" class="box">

<div class="in">

<h2>QR Code Generator</h2>
<form action="/generate-qr" method="POST">
<!-- <label for="qr-data">:</label> -->
<input type="text" id="qr-data" name="data" placeholder="Enter Text or Link" class="inputbar" required>
<button type="submit">Generate QR Code</button>
</form>
</div>

</section>

<!-- PDF Merger -->
<section id="pdf-merger" class="box">

<div class="in">

<h2>PDF Merger</h2>
<form action="/merge-pdfs" method="POST" enctype="multipart/form-data">
<div class="under">
<label for="pdf-files">Select PDFs to Merge:</label>
<input type="file" id="pdf-files" name="pdfs" placeholder="" class="inputbar" multiple required>
</div>


<button type="submit">Merge PDFs</button>
</form>
</div>

</section>

<!-- Image Compressor -->
<section id="image-compressor" class="box">

<div class="in">

<h2>Image Compressor</h2>
<form action="/compress-image" method="POST" enctype="multipart/form-data">

<div class="under">
```

```html
<label for="image-file">Select Image to Compress:</label>
<input type="file" id="image-file" name="image" placeholder="" class="inputbar" required>

</div>

<button type="submit">Compress Image</button>
</form>
</div>

</section>

<!-- Barcode Generator -->
<section id="barcode-generator" class="box">

<div class="in">

<h2>Barcode Generator</h2>
<form action="/generate-barcode" method="POST">
<!-- <label for="barcode-code">Enter Code:</label> -->
<input type="text" id="barcode-code" name="code" placeholder="Enter Code" class="inputbar" required>
<div class="under">

<label for="barcode-format">Select Format:</label>
<select id="barcode-format" name="format">
<option value="ean13">EAN-13</option>
<option value="ean8">EAN-8</option>
<option value="upca">UPC-A</option>
</select>

</div>

<button type="submit">Generate Barcode</button>
</form>
</div>

</section>

<!-- Instagram Reel Downloader -->
<section id="instagram-downloader" class="box">

<div class="in">

<h2>Instagram Reel Downloader</h2>
<form action="/download-instagram" method="POST">
<!-- <label for="insta-url">:</label> -->
<input type="text" id="insta-url" name="url" placeholder="Enter Instagram Reel URL" class="inputbar" required>
<button type="submit">Download Reel</button>
```

```html
</form>
</div>

</section>

<!-- Images to PDF Converter -->
<section id="images-to-pdf" class="box">

<div class="in">

<h2>Images to PDF Converter</h2>
<form action="/convert-images-to-pdf" method="POST" enctype="multipart/form-data">
<div class="under">
<label for="image-files">Select Images:</label>
<input type="file" id="image-files" name="images" placeholder="" class="inputbar" multiple
required>
</div>
<button type="submit">Convert to PDF</button>
</form>
</div>

</section>

<!-- YouTube to MP3 Downloader -->
<section id="yt-mp3" class="box">

<div class="in">

<h2>YouTube to MP3 Downloader</h2>
<form action="/download-mp3" method="POST">
<!-- <label for="yt-mp3-url">Enter YouTube Video URL:</label> -->
<input type="text" id="yt-mp3-url" name="url" placeholder="Enter YouTube Video URL"
class="inputbar" required>
<button type="submit">Download as MP3</button>
</form>

</div>

</section>

</div>




</body>
</html>
```

STYLE.CSS

```
*{

margin: 0%;
padding: 0%;
}

#arrowup{

width: 36px;

}
.arrow{

/* color: rgb(221, 5, 5); */
background-color: white;
height: 60px;
width: 60px;
border-radius: 2.1rem;
position: fixed;
right: 13px;
top: 530px;

display: flex;
align-items: center;
justify-content: center;
}


.arrow:hover{

border: 2px solid black;
height: 62px;
width: 62px;

}


.upperdiv{

/* background-color: #00DBDE;
background-image: linear-gradient(90deg,#FC00FF 0%, #00DBDE 100%); */
```

```
background-color: rgba(0, 0, 0, 0.925);
/* transition-duration: 2s;

animation: change 2s infinite alternate; */
}

/* @keyframes change {
100%{

background-color: #F4D03F;
background-image: linear-gradient(132deg, #F4D03F 0%, #16A085 100%);


}} */


#h1head{
color: white;
}

@keyframes heading {

100%{
text-shadow: 0 0 60px rgba(9, 5, 243, 0.699);

}
}


header{

display:grid;
justify-items: center;
font-family: 'Courier New', Courier, monospace;

padding: 2rem;

padding-top: 1rem;
/* margin: 2rem; */

font-size: 1.8rem;
/* color: white; */

color: white;

text-shadow: 1px 1px 2px black;
}

p{
```

```
font-weight: bold;
/* box-shadow: 20px 20px 60px rgba(0, 0, 0, 0.438), inset -20px -20px 60px white; */
}


.headdiv{
display: flex;
}

#headerimg{
height: 3rem;
width: 3rem;
padding-right: 1.5rem;
padding-top: 0.6rem;
}


/* nav{

display: flex;
justify-content: space-around;
margin: 2rem;
font-size: 1.5rem;
text-align: center;
} */

.navclass{
display: grid;
justify-content: center;
/* background: rgba(104, 98, 98, 0.089); */

box-sizing: border-box;

padding-bottom: 2rem;
/* background-color: black; */
}

nav{
display: grid;
grid-template-columns: repeat(3,1fr);
column-gap: 3rem;
row-gap: 1.2rem;
background-color:rgba(255, 255, 255, 0.171);

padding: 2rem;
border-radius: 3rem;
backdrop-filter: blur(10px);
border: 1px solid black;
}
```

```
.link{
padding-bottom: 2rem;
/* margin: 1.1rem; */
/* border: 2px solid black; */
text-decoration: none;
color: white;
font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
font-size: 1.2rem;
text-shadow: 1px 1px 2px black;

}


.main{
background-color: #00DBDE;
background-image: linear-gradient(90deg, #00DBDE 0%, #FC00FF 100%);
display: grid;
/* grid-template-columns: repeat(3,1fr) ; */

/* for text */
justify-content: center;

}



.box{

display: grid;
/* grid-template-columns: repeat(3,1fr) ; */

/* for text */
justify-content: center;

height: 550px;
padding: 10px;
position: relative;

margin: 2rem;
/* margin-top: 3rem; margin: 2rem; */
/* background-color: rgb(255, 255, 255); */
overflow: hidden;
/* text-shadow: 0 0 60px rgb(216, 14, 14); */
animation: heading 1s infinite alternate;
}
.in{
width: 70rem;
background-color: rgb(0, 0, 0);
border: 2px solid black;
color: rgb(255, 255, 255);
position: relative;
z-index: 99;

padding: 2rem;

/* display: grid; */
text-align: center;
font-size: 2rem;
font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;

border-radius: 2rem;

transition: all 2s;

}

.in:hover{

background-color: rgb(255, 255, 255);
color: rgb(0, 0, 0);


}

/*
.in:hover-box::after{

background-color: black;
}; */
.box::after{
content: "";
position:absolute;
top: 50%;
left: 50%;
transform: translate(-50%,-50%) rotate(45deg);
height:1350px;
width: 200px;
background-color: rgb(255, 255, 255);

animation: move 1.5s infinite;
/* border: 2px solid black; */
}
@keyframes move {
100%{
transform: translate(-50%,-50%) rotate(225deg);

}}
```

```css
form{
display: grid;
/* justify-items: center; */

margin: 2rem;
padding: 2rem;
/* justify-content: space-around; */
/* width: 100%; */

}

.navbox{

height: 7rem;
width: 18rem;
border-radius: 2rem;
display: grid;
justify-items: center;
box-shadow: 20px 20px 60px rgba(0, 0, 0, 0.438), inset -20px -20px 60px rgba(255, 255, 255, 0.74);
border: 1px solid rgba(105, 95, 95, 0.11);
}

.navbox:hover{
box-shadow: none;
border: 1px soli black ;
}

.othergrid{
display: grid;
grid-template-columns: repeat(2,1fr);
}


.imgclass{
height: 2rem;
width: 2rem;
margin-top: 1.2rem;
}



#imgtag1{
background-color: #4a90e2;
}
#imgtag2{
background-color: #FF6289;
}
#imgtag3{
background-color: #fcbf58;
}
#imgtag4{
background-color: #19dbe2;
}
#imgtag5{
background-color: #77b05d;
}
#imgtag6{
background-color: #7d78b1;
}
#imgtag7{
background-color: #1f7eaa;
}
#imgtag8{
background-color: #a73b7d;
}
#imgtag9{
background-color: #b84343;
}




.inputbar{
padding: 1rem;
font-size: 1.1rem;
margin: 2rem;
border-radius: 1rem;


}

button{
margin: 4rem;
margin-left: 25rem;
height: 3rem;
width: 11rem;
border-radius: 1.2rem;
font-size: 1rem;
color: white;
background-color: black;
border: 1px solid white;
font-family: Georgia, 'Times New Roman', Times, serif;}
```

APP.PY

```
from flask import Flask, render_template, request, send_file, jsonify

import yt_dlp
import qrcode
import os
from io import BytesIO
from PyPDF2 import PdfMerger
from PIL import Image
import barcode
from barcode.writer import ImageWriter
import instaloader

app = Flask(__name__)

# Path where videos will be temporarily saved
DOWNLOAD_FOLDER = os.path.join(os.path.expanduser("~"), "Downloads/website")
download_progress = 0 # Global variable to store download progress percentage

# Specify the path to FFmpeg
FFMPEG_LOCATION = 'C:/ffmpeg/' # Ensure this is the correct path to your FFmpeg installation


# YouTube Downloader (MP4 and MP3)
@app.route('/')
def index():
return render_template('index.html')


@app.route('/download', methods=['POST'])
def download_video():
global download_progress
url = request.form['url']
download_progress = 0 # Reset progress at the start of the download

def progress_hook(d):
global download_progress
if d['status'] == 'downloading':
total_bytes = d.get('total_bytes', 0)
downloaded_bytes = d.get('downloaded_bytes', 0)
if total_bytes > 0:
download_progress = int(downloaded_bytes / total_bytes * 100)
```

```
try:
ydl_opts = {
'outtmpl': os.path.join(DOWNLOAD_FOLDER, '%(title)s.%(ext)s'),
'progress_hooks': [progress_hook],
'ffmpeg_location': FFMPEG_LOCATION # Set the correct FFmpeg location
}

with yt_dlp.YoutubeDL(ydl_opts) as ydl:
info_dict = ydl.extract_info(url, download=True)
video_title = info_dict.get('title', None)
video_ext = info_dict.get('ext', None)
file_path = os.path.join(DOWNLOAD_FOLDER, f"{video_title}.{video_ext}")

return send_file(file_path, as_attachment=True)

except Exception as e:
return jsonify({'error': str(e)}), 500


@app.route('/progress')
def get_progress():
global download_progress
return jsonify({'progress': download_progress})


@app.route('/download-mp3', methods=['POST'])
def download_mp3():
url = request.form['url']

ydl_opts = {
'format': 'bestaudio/best',
'postprocessors': [{
'key': 'FFmpegExtractAudio',
'preferredcodec': 'mp3',
'preferredquality': '192',
}],
'outtmpl': os.path.join(DOWNLOAD_FOLDER, '%(title)s.%(ext)s'),
'ffmpeg_location': FFMPEG_LOCATION # Set the correct FFmpeg location
}

try:
with yt_dlp.YoutubeDL(ydl_opts) as ydl:
ydl.download([url])
info_dict = ydl.extract_info(url, download=True)
file_path = os.path.join(DOWNLOAD_FOLDER, f"{info_dict['title']}.mp3")

return send_file(file_path, as_attachment=True)
```

```
except Exception as e:
return jsonify({'error': str(e)}), 500


# QR Code Generator
@app.route('/qr-scanner')
def qr_scanner():
return render_template('qr_scanner.html')


@app.route('/generate-qr', methods=['POST'])
def generate_qr():
data = request.form['data']
qr_img = qrcode.make(data)

img_io = BytesIO()
qr_img.save(img_io, 'PNG')
img_io.seek(0)

return send_file(img_io, mimetype='image/png', as_attachment=True,
download_name='qrcode.png')


# PDF Merger
@app.route('/pdf-merger')
def pdf_merger():
return render_template('pdf_merger.html')


@app.route('/merge-pdfs', methods=['POST'])
def merge_pdfs():
pdf_files = request.files.getlist('pdfs')
merger = PdfMerger()

for pdf in pdf_files:
merger.append(pdf)

merged_pdf = BytesIO()
merger.write(merged_pdf)
merged_pdf.seek(0)

return send_file(merged_pdf, as_attachment=True, download_name='merged.pdf',
mimetype='application/pdf')


# Image Compressor
@app.route('/image-compressor')
def image_compressor():
return render_template('image_compressor.html')
```

```
@app.route('/compress-image', methods=['POST'])
def compress_image():
image = Image.open(request.files['image'])
img_io = BytesIO()

# Compress image to reduce quality
image.save(img_io, format='JPEG', quality=20)
img_io.seek(0)

return send_file(img_io, as_attachment=True, download_name='compressed_image.jpg',
mimetype='image/jpeg')


# Barcode Generator
@app.route('/barcode-generator')
def barcode_generator():
return render_template('barcode_generator.html')


@app.route('/generate-barcode', methods=['POST'])
def generate_barcode():
try:
# Extract form data
code = request.form['code']
barcode_format = request.form.get('format', 'ean13') # Default to 'ean13'

# Log inputs for debugging
print(f"Generating barcode. Format: {barcode_format}, Code: {code}")

# Check if the barcode format is valid
try:
BARCODE_CLASS = barcode.get_barcode_class(barcode_format)
except barcode.errors.BarcodeNotFoundError:
return jsonify({'error': f'Invalid barcode format: {barcode_format}'}), 400

# Validate the code for EAN-13 format (must be 13 digits)
if barcode_format == 'ean13' and not (code.isdigit() and len(code) == 13):
return jsonify({'error': 'EAN-13 barcode requires a 13-digit numeric code.'}), 400

# Generate the barcode image
barcode_obj = BARCODE_CLASS(code, writer=ImageWriter())
img_io = BytesIO()
barcode_obj.write(img_io)
img_io.seek(0)

# Return the generated barcode image as a PNG file
return send_file(img_io, as_attachment=True, download_name='barcode.png',
mimetype='image/png')

except Exception as e:
# Log any errors
print(f"Error generating barcode: {str(e)}")
return jsonify({'error': str(e)}), 500


# Instagram Reel Downloader
@app.route('/instagram-downloader')
def instagram_downloader():
return render_template('instagram_downloader.html')


@app.route('/download-instagram', methods=['POST'])
def download_instagram():
url = request.form['url']

loader = instaloader.Instaloader()
shortcode = url.split("/")[-2]

try:
loader.download_post(instaloader.Post.from_shortcode(loader.context, shortcode),
target="downloads")
return "Reel downloaded successfully!"
except Exception as e:
return str(e)


# Images to PDF Converter
@app.route('/images-to-pdf')
def images_to_pdf():
return render_template('images_to_pdf.html')


@app.route('/convert-images-to-pdf', methods=['POST'])
def convert_images_to_pdf():
image_files = request.files.getlist('images')
img_list = [Image.open(img) for img in image_files]

pdf_io = BytesIO()
img_list[0].save(pdf_io, save_all=True, append_images=img_list[1:], format='PDF')
pdf_io.seek(0)

return send_file(pdf_io, as_attachment=True, download_name='images_to_pdf.pdf',
mimetype='application/pdf')
if __name__ == '__main__':
app.run(debug=True)
```

Technologies and Tools Used

The **ALL-IN-ONER** project leverages a variety of modern technologies and libraries to deliver its diverse functionality:

1. **Flask (Python Framework)**:
   - Flask serves as the backbone of the web application, providing a lightweight and flexible framework for handling HTTP requests and rendering dynamic pages. Its simplicity allows for rapid development and easy scalability.
2. **yt-dlp**:
   - This powerful command-line tool is utilized for downloading videos and extracting audio from platforms like YouTube. With its rich feature set, `yt-dlp` allows users to easily obtain media content in various formats.
3. **qrcode**:
   - A dedicated Python library for generating QR codes from input data. This tool enables the application to create scannable codes, enhancing user interaction and accessibility.
4. **PyPDF2**:
   - Used for merging multiple PDF files into a single document. This library facilitates the handling of PDF files, allowing users to compile and manage their documents efficiently.

5. **Pillow (PIL)**:
    - An imaging library that allows for comprehensive handling and manipulation of image files. With Pillow, the application can compress images and convert formats, enhancing user experience with visual content.
6. **python-barcode**:
    - A library designed for generating various types of barcodes, including EAN-13, UPC-A, and EAN-8. This tool enables users to create barcodes based on input data, supporting various applications.
7. **Instaloader**:
    - A specialized tool that allows users to download Instagram media, such as posts and reels, simply by providing the media URL. Instaloader simplifies the process of accessing content from Instagram, making it easier for users to save and share media.

Conclusion

These technologies and tools collectively enable the **ALL-IN-ONER** project to provide a seamless and efficient user experience, catering to a wide range of functionalities from video downloading to QR code generation and media handling. By leveraging modern libraries and frameworks, the project showcases a robust architecture capable of meeting user needs.

Future Improvements

While the **ALL-IN-ONER** platform is already robust and feature-rich, there are several promising avenues for future enhancements that could further increase its utility and user engagement:

1. **Adding More File Conversion Tools**:

    - Expanding the range of conversion tools to include features such as video-to-GIF conversion, document format converters (e.g., Word to PDF, Excel to CSV), and audio format converters will significantly broaden the platform's capabilities. This enhancement would cater to diverse user needs and preferences.
    -

2. **User Authentication and Cloud Storage**:

    - Implementing user authentication features would allow users to create accounts, save their work, and access previously generated files easily. This personalization could enhance user satisfaction and retention.
    - Integrating cloud storage solutions would enable users to store their downloaded or generated files directly in the cloud, providing easy access from multiple devices and reducing local storage dependency.
    -

3. **Support for More Media Platforms**:

    - Expanding the downloader's capabilities to support additional popular media platforms such as Facebook, Vimeo, Twitter, and TikTok will make the tool even more versatile. This improvement would attract a broader user base and meet the growing demand for content from various sources.
    -

4. **Enhanced User Interface and Experience**:

   - Continuous improvement of the user interface (UI) and user experience (UX) design based on user feedback and usability testing will ensure that the platform remains intuitive and easy to navigate. Incorporating responsive design principles will also improve accessibility across devices.
   -

5. **Advanced Features for Video Downloading**:

   - Introducing options for downloading playlists, channels, or bulk media downloads from platforms like YouTube will enhance functionality. Adding features like video trimming, subtitle download, and format selection would also improve user experience.
   -

6. **Integration of Machine Learning**:

   - Exploring machine learning techniques to provide personalized recommendations for users based on their usage patterns could enhance user engagement. For example, suggesting relevant tools or media based on past activities could create a more tailored experience.

Conclusion

The **ALL-IN-ONER** project represents a significant step forward in simplifying multimedia processing tasks for users by integrating multiple functionalities into one cohesive platform. This web application not only enhances user convenience but also streamlines workflows that would typically require multiple tools.

By providing features such as YouTube video downloading, QR code generation, PDF merging, image compression, barcode generation, Instagram reel downloading, and images-to-PDF conversion, **ALL-IN-ONER** caters to a wide audience, from students to professionals, enabling them to accomplish a variety of tasks efficiently. The use of Flask as the backend framework ensures a robust and scalable application that can handle user requests seamlessly.

Looking ahead, there are numerous opportunities for future improvements. The addition of user authentication and cloud storage options can enhance user experience and data management. Furthermore, expanding support for additional media platforms and file conversion tools will make the application even more versatile and valuable.

In summary, **ALL-IN-ONER** is a forward-thinking project that addresses the growing need for comprehensive multimedia processing solutions. By continually evolving and integrating user feedback, it has the potential to become an indispensable tool for a broad spectrum of users in today's digital landscape.