

URL to PDF Converter

A full-stack web application that crawls a given website URL and converts its pages into downloadable PDF files. Each crawled page is converted into an individual PDF with preview support, and all PDFs can also be merged into a single file.

This project is built for learning, experimentation, and academic use.

Key Features

- Convert any website URL into PDF
 - Automatically crawls same-domain pages
 - Generates:
 - Individual page PDFs
 - One merged PDF
 - Live PDF preview using iframe
 - Download individual PDFs or merged PDF
 - Automatically fixes URLs without `http://` or `https://`
 - Handles static websites reliably
-

How the System Works

1. User enters a website URL in the frontend
 2. Frontend sends URL to backend using Fetch API
 3. Backend:
 4. Normalizes the URL
 5. Crawls pages within the same domain
 6. Uses Puppeteer to render pages
 7. Generates PDFs
 8. Merges PDFs into a single file
 9. Frontend:
 10. Displays preview of each PDF
 11. Provides download buttons
-

Tech Stack

Frontend

- HTML
- CSS
- JavaScript (Fetch API)

Backend

- Node.js
 - Express.js
 - Puppeteer
 - Axios
 - Cheerio
 - pdf-lib
-



Project Structure

```
URL2PDF/  
├── backend/  
│   ├── crawler.js      # Crawls website pages  
│   ├── pdf.js          # Generates PDFs using Puppeteer  
│   ├── merge.js        # Merges multiple PDFs  
│   ├── server.js       # Express server & API  
│   ├── output/         # Generated PDF files  
│   ├── package.json  
│   └── node_modules/  
├── frontend/  
│   └── index.html      # User Interface  
└── README.md
```

Installation & Setup



Backend Setup

```
cd backend  
npm install  
node server.js
```

Server will run on:

```
http://localhost:3000
```

Frontend Setup

Open this file in a browser:

```
frontend/index.html
```

(Optional: use VS Code Live Server)

Important Behavior Notes

PDF Preview

- Preview is rendered using `<iframe>`
- Links inside preview remain clickable (default browser behavior)

Crawling

- Crawls **same-domain URLs only**
- Page count is limited for safety
- Dynamic content may not fully render

PDF Layout

- Layout depends on website CSS
 - Some margins/spacing may differ from browser view
 - Extra pages can occur due to page height overflow
-

Possible Enhancements

- User-defined crawl depth
 - Disable links inside PDF preview
 - Remove blank PDF pages automatically
 - Add progress indicator
 - Support login-protected pages
 - Deploy backend using Docker or cloud services
-

Author

Chaitanya Falari

Built as a learning & academic project.
