COS 498

Assignment 3: Secure PDF Server

15pts

Due: Nov 25th (Tuesday), 11:59pm

1 Assignment Description

For this assignment, you will create an Express + Nginx proxy manager server that serves a front-facing website with PDF document management capabilities. You will implement custom modules to handle routing, PDF discovery, and PDF validation. Additionally, you will set up your site with a domain name and HTTPS support.

You're free to pick the topic of your website, but it must be related to the PDF documents you are hosting.

2 Required Components

2.1 Front-Facing Website

- Create a professional front-facing website that serves as the main interface
- The website should provide relevant content related to the PDF documents you are hosting
- The website should display a list of available PDF documents
- Ensure the website is well-designed and user-friendly

2.2 PDF Document Management

- Store PDF documents in a designated folder (e.g., `pdfs/` or `documents/`)
- The server must be able to serve PDF documents to clients
- **Do not use the static file middleware to serve the PDFs.** Use the `sendFile()` function from Express to send the files to the client once you confirmed that the client can access the file.
- PDFs should be accessible via appropriate URLs (e.g., `/pdfs/document-name.pdf`)
- Implement a system to store metadata about PDFs (see Database/Storage Requirements below)
- Have at least 3 PDFs in your folder for testing.

2.3 Domain Name and HTTPS

- Obtain and configure a domain name for your website
- Set up SSL/TLS certificates to enable HTTPS
- Ensure your site is accessible via `https://yourdomain.com`
- All HTTP traffic should redirect to HTTPS (optional but recommended)

2.4 Custom Modules

You must create your own custom modules (separate JavaScript files) to handle the following functionality:

2.4.1 Routing Module

- Create a custom routing module that handles URL routing for your application
- The module should map URLs to appropriate handlers or static files
- Implement route definitions for your main pages and PDF endpoints
- Handle 404 errors for routes that don't exist

2.4.2 PDF Discovery Module

- Create a module that searches for available PDF documents in a designated folder
- The module should scan the folder and create a list of available PDFs
- This list should be used to display available PDFs on your website
- The module should handle file system operations to read directory contents
- Consider caching the PDF list to avoid repeated file system reads

2.4.3 PDF Validation Module

- Create a module that checks if a requested PDF document exists
- Before serving a PDF, validate that the file exists in the designated folder
- Return appropriate error responses (e.g., 404) if the PDF doesn't exist
- Only allow access to PDFs within the designated folder

3 Database/Storage Requirements

Decide how to store information about PDFs so that clients see more than just the PDF filename. You have two options:

3.1 Option 1: SQLite3 Database

- Use SQLite3 to store metadata about PDF documents
- Create a table that stores information such as:
  - PDF filename + path
  - Display title/name
  - Description
- Implement functions to query and update the database
- Display this metadata on your website when listing PDFs

3.2 Option 2: Text Files

- Create additional text files (e.g., JSON, or plain text) that describe the PDFs
- Each PDF could have a corresponding metadata file (e.g., `document.pdf` and `document.pdf.json`)
- Or use a single metadata file that contains information about all PDFs
- Parse these files to display meaningful information about PDFs on your website
- Include at minimum: display title/name and description for each PDF

4 Website Features

Your front-facing website should include:

- A homepage with relevant content related to your PDF collection
- A page or section that displays a list of available PDF documents
- Each PDF in the list should show:
  - A meaningful title (not just the filename)
  - A description
  - A link to view/download the PDF
  - Any other metadata you've stored
- Navigation between different sections of your site
- Professional styling and layout (you can use AI for this)

5 Testing and Documentation

5.1 Testing Requirements

- Test that your server starts successfully
- Verify that PDFs are discovered correctly
- Test PDF validation (both existing and non-existing PDFs)
- Verify routing works for all pages
- Test HTTPS connection
- Verify domain name resolves correctly
- Ensure PDF list displays correctly with metadata

5.2 Documentation

- Include comments in your code explaining your modules
- Document your database schema (if using SQLite3) or metadata file format (if using text files)
- Explain your routing structure
- Document any setup steps required (domain configuration, SSL certificate setup, etc.)

6 Point Distribution

Component	**Points**
Front-facing website with PDF list display	3pts
Domain name configured and working	1pt
README.md file has all documentation requirements	2pts
HTTPS setup and working	2pts
Custom routing module	2pts
Custom PDF discovery/validation module	3pts
PDF metadata storage (SQLite3 or text files)	2pts
No comments at the top of server file and modules	-2pts
Very few or no comments for the different functions	-2pts
Site makes my eyes bleed	-2pts
Total	**15pts**

7 Submission Guidelines

- Submit the URL to your hosted website (with domain name) through Brightspace
- Submit the github repository link through Brightspace (make sure it's public)
- Ensure your site is accessible via HTTPS
- In your README.md include a brief description of your project and how to run it, as well as all your documentation requirements.
- Test your server and website thoroughly before submission