

Full stack project

Build a File sharing web app like Dropbox

This project will help you in getting a full understanding of Python Flask web framework, HTML, CSS, Javascript and Bootstrap UI framework and will hugely boost your profile as a developer.

Enrol in the project here : <https://flinkhub.com/project/fullstack-project>

What you'll learn

- How to work like a Full-stack developer at a tech company
- How web applications are built
- HTTP protocol in detail
- Python Flask web framework
- Bootstrap UI framework
- MongoDB
- Building Authentication Flows
- Handling File uploads and downloads

Project Stages

Stage 1: Brush up on fundamentals

- Learn Python, HTML, CSS, JS, Bootstrap fundamentals
- Understanding MongoDB and how data is stored and retrieved

Stage 2: Login & Sign up

- Set up the data structure
- Understand HTTP sessions
- Implement a secure authentication flow

Stage 3: Get started with Files

- Create UI
- Show list of uploaded files in the UI
- Handling file uploads

Stage 4: File Sharing

- Make files shareable via permalinks
- Allow downloading of uploaded files

About Instructor

I'm Ananth SNC. I started my software development journey as a freelance programmer 10 years ago, while I was studying at IIT Kharagpur. During this time, I have learnt many programming languages, built web and Mobile apps for 200+ companies and founded 3 companies on my own.

Who should enroll

Anyone who has basic programming knowledge and is looking to add a real project to their profile for full stack or backend developer roles.

Time Commitment

100 - 120 hours in total

Project Specifications

Background

The goal of this project is for you (the student) to learn full stack development by building an entire web application. When you make a web application like this all by yourself, you will fully understand how all web applications work. You will have a good understanding of the HTTP Protocol, handling data in databases, building user interfaces (UIs) and connecting these UIs with real data.

Though you will be using Python language in this project, these concepts are the same for any web app, so work on backend in any another language using the same concepts. We chose Python for this project because of the easy learning curve, which means you will be able to understand it and implement your project very quickly.

For the frontend, you will be using Bootstrap UI framework. During the project, you will go through various UI components available in Bootstrap framework so that you can use the right components while developing the UI.

Problem Statement

In this project, you will be building a file sharing web application like [Dropbox.com](https://www.dropbox.com) or Google Drive

Problem Details

The web app must be secured through a login / sign up feature. Users must sign up for a new account if they do not have an account. And they must login before accessing the file sharing features of the application.

After logging in, users should see the list of files they have uploaded previously. They should also be able to upload new files through a Drag & Drop UI.

We will be limiting the type of files that can be uploaded by users. The allowed file types are: JPG, GIF, PNG, DOC, DOCX, XLS, XLSX, PPT, PPTX, PDF, CSV - only. If a user uploads a file that is not among these, the system should reject the file. (Note that we will not be allowing audio and video files in this system.)

The maximum allowed file size should be 20 MB and total upload size per user is 300 MB.

After uploading a file, each file should have a permalink, which can be shared by the user publicly. Any logged in user can then download the file from this link. Please note that even though the link can be shared publicly, the other user must be signed in (using their account) to access and download the file. The files should not be accessible publicly to prevent non-authenticated users from downloading them.

The system should track downloads i.e. which user has downloaded which file.

Starter UI wireframes for Login, Signup, Files List, Upload dialog, Share dialog & Download File pages are given below.

My Box

Login to your account

Don't have an account? [Sign up](#)

My Box

Create a new account

Already have an account? [Login Now](#)

My Box

user@email.com Logout

Uploaded Files

Upload

File Name	Size	Uploaded		
My Document 1.docx	3.6 MB	6 days ago	Share	Delete
My Photo.png	784 KB	8 days ago	Share	Delete
Another file.pdf	15.3 MB	12 days ago	Share	Delete
Abstract Wallpaper Image.png	1.3 MB	17 days ago	Share	Delete
Another important file.pdf	2.7 MB	21 days ago	Share	Delete

My Box

user@email.com Logout

Uploaded Files

Upload

Upload New File x

Click below to upload a file or Drag and drop the file you want to upload to the below container.

Click to Upload or Drag & Drop File

Upload

My Box user@email.com Logout

Uploaded Files Upload

Share File x

You can copy this link to share **Another file.pdf**

<http://localhost:5000/file/5f36db284d6389838c86c31b/another-file-pdf>

Copy to Clipboard

File N
My D
My P
Anoth
Abstra
Anoth

My Box user@email.com Logout

This file has been shared with you.

My Document 1.docx	3.6 MB	6 days ago	Download
--------------------	--------	------------	-----------------------

Database

For this project, you must use MongoDB as the database. Files will be stored on the disk in a set folder.

Data Structure

There will be multiple collections in this application, and more collections may be added at a later stage. Please note that every document will have a unique `_id` ObjectId field. If you would like to add any other useful fields to the data structure, feel free to add it.

1. **Users** – This collection will store users data
 - a. Email – The email address of the user
 - b. Password – Password of user encrypted using the SHA-256 algorithm (and using a salt phrase)
 - c. Name – Full name of the user
 - d. Last Login date – Date at which the user last logged in
 - e. Created at – Date at which this user signed up
 - f. Updated at – Date at which this user object was last updated

2. **User_Tokens** – This collection will store authentication tokens
 - a. User ID – The ID of user
 - b. Session hash – Random SHA256 encrypted string
 - c. Created at – Date at which this token is created

3. **Files** – This collection will store file details
 - a. User ID – The ID of user who uploaded the file
 - b. Original file name – Name of the file when it was uploaded
 - c. File type – Extension of the file
 - d. File hash – SHA256 checksum of file
 - e. File size – Size of the file in bytes
 - f. File path – Path to the uploaded file stored on disk
 - g. Is Active – Whether this file is still active (true/false, will be false if file is deleted by user)
 - h. Created at – Date at which this file was uploaded

4. **File_Downloads** – This collection will store download details
 - a. User ID – The ID of user who downloaded the file

- b. File ID – The ID of the file
- c. Created at – Date at which this download happened

Tools & Resources

- Python 3 Interpreter – You must do this project using Python 3.8
<https://www.python.org/downloads/>
- IDE / Code Editor – As per your choice / PyCharm / VS Code
<https://www.jetbrains.com/pycharm/>
<https://code.visualstudio.com/>
- Robo 3T – To browse MongoDB database contents
<https://robomongo.org/download>
- Flask web framework
<https://flask.palletsprojects.com/en/1.1.x/>
- Bootstrap UI framework
<https://getbootstrap.com/>
- Pymongo
<https://pymongo.readthedocs.io/en/stable/tutorial.html>
- Useful tutorial videos
Learn Flask Freecodecamp - https://www.youtube.com/watch?v=Z1RJmh_OqeA
Using Flask and Pymongo together - <https://www.youtube.com/watch?v=HyDACIfdPs0>
Using Virtualenv - <https://www.youtube.com/watch?v=1v46QeHtvJ4>

Open source modules

In this project, you must use Flask web framework for the backend, Bootstrap UI framework for the frontend, and MongoDB for database. You are free to use any other open source modules that can help you in building this application.

Submission Details

You must submit your code as a public git repository on github.com . The name of the repository should be “file-sharing-app”. After pushing the code, send me the link to your github

repository in DM to @a1ananth on Telegram (<https://t.me/a1ananth>). There is no deadline for this project.

Notes

- Application configuration must be present in a separate `cfg.py` file
- You must use the logging module to print debug, info, warning, error logs at different stages in the application
- You must use `virtualenv` to create a virtual environment for the app and add a `requirements.txt` file for dependencies
- You must build a secure (not easily hackable) authentication flow using tokens. You can study about “token based authentication” for more information
- Certificate will be provided ONLY to people who submit working code that satisfies ALL problem details
- Do not check in any sensitive information like usernames and passwords into your Github repositories