

Mini Project Presentation

PDFly An Online PDF Utility

Submitted by:
Shraddha Singh



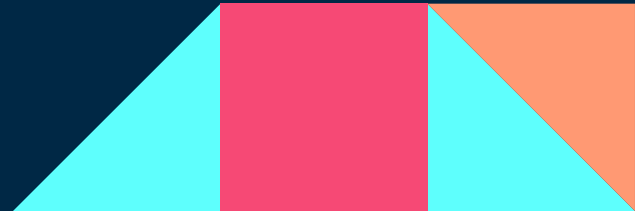
Submitted to:
Dr Anurag Sewak

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RAJKIYA ENGINEERING COLLEGE, SONBHADRA, UTTAR PRADESH

Introduction

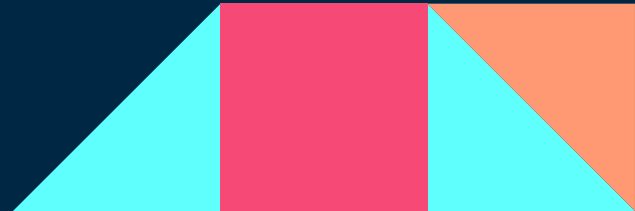
PDFs are an integral part of common lives of working professionals as well as students as they are one of the most preferred methods of exchanging data containing various components such as text-formattings, inline images, fonts, vector graphics, etc.

Thus, there arises a need for a system that allows users to manipulate and perform various functions on PDFs as per their needs. Thus, a system or interface that helps user to have efficient solutions at reasonable price with a lot of functionality at single place is highly appreciated.



OUTLINE

- ❏ Scope
- ❏ Advantages
- ❏ Objective
- ❏ Working and Background
- ❏ Algorithm
- ❏ Interface
- ❏ References



SCOPE

❏ Education and Academics

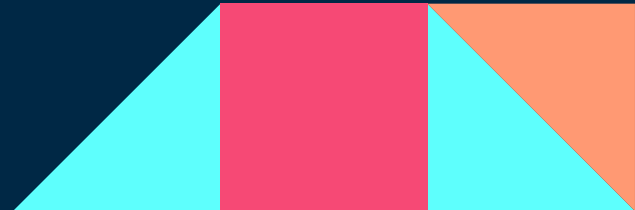
- ❏ Students as well as teachers require online pdf utilities to deal with pdfs as they are the most preferred method of sending and receiving any data whether it be homework or assignment.

❏ Offices and work Environments

- ❏ Working Professionals have a lot of work that depends on pdfs such as report submissions and task allotments as well as official documentations in the offices.

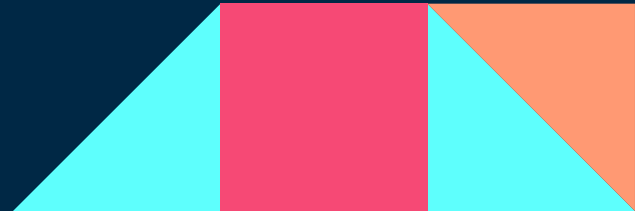
❏ Media

- ❏ Ebooks and various other form of media are served in form of PDFs which makes online pdf utilities a must for such platforms and their users.

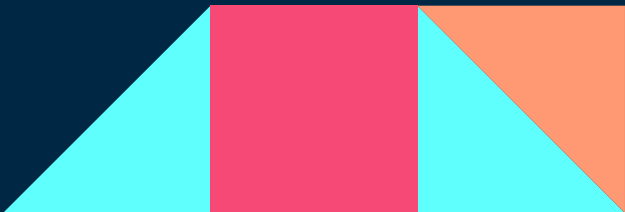


Advantages of PDFly

- ❑ It will allow students to have the required functionalities at their hands at one place and that too for a nominal fee.
- ❑ Working Professionals can have one subscription rather than having individual ones in order to minimise their expenses.
- ❑ The implementation is quite straightforward and the functionalities can be expanded as per need since the code is quite simple.
- ❑ The interface is really simple and anyone can use it without having to worry about a lot of things.



Objective of the Project

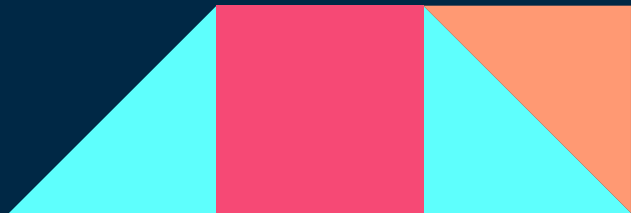
- ❑ To provide various functionalities related to PDF Manipulations and conversions at one place.
 - ❑ To provide an interface that is really simple and doesn't require any signup to use it.
 - ❑ To help companies and institutes to minimize costs by purchasing only one subscription and allowing access to anyone with the deployed link for that organization.
 - ❑ To make manipulating and converting PDFs a hassle free and free with the help of this utility.
- 

Working And Background

This online PDF Utility uses ExpressJS in the backend and utilises EJS and CSS3 to serve the front end. The file is uploaded to the backend with the help of “Multer” a third party node package.

Then, the server sends the uploaded file to the PDFTRON API where it performs the actions stated in the functions that are defined in the backend more precisely a file named app.js. These function are asynchronous in nature and for that we have used Asynchronous JS.

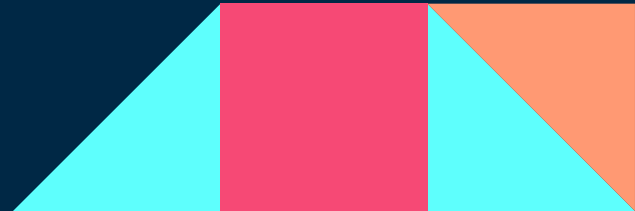
When the API returns the generated file it is saved in the backend with the help of path defined by us.



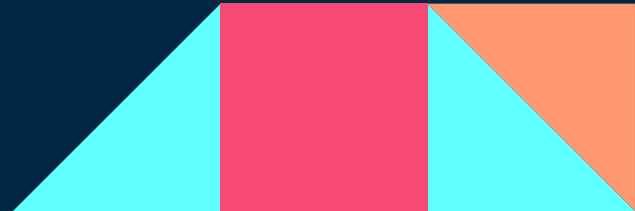
Then the server is made to redirect the client to the download page where he/she can download the file he has wished for from the beginning.

Components of the Project

- ❑ Frontend: The front end is designed with the help of EJS and CSS3, we have used connect-flash node package to flash messages on the screen on redirects. The EJS files are served with the help of express-ejs-layouts to the user's screen. The front end consists of all the forms required to gather file that needs to be operated upon and various other requirements.



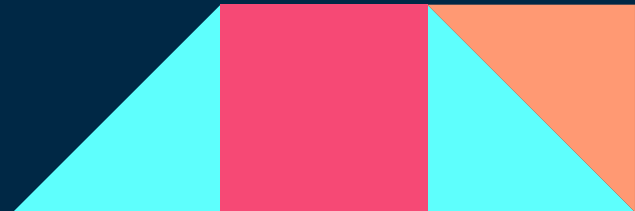
- ❑ Backend: The back end of the project is build on Node.JS with the help of its framework ExpressJS. The backend is responsible for serving appropriate file as per the route requested by the user. It gets the data from the user and interacts with the API to perform operations on that data. It consists of the various node modules such as:
 - ❑ multer: It is a third party node package that enables us to upload files of various formats to our server, in our case we have configured Multer to use local disk storage for that purpose.



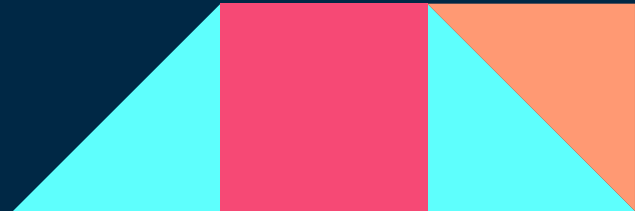
- ❑ Express-session: connect-flash depends on sessions, for that we have set up express-session in our backend.
- ❑ Nodemon: It is a really helpful module that restarts the server every time any change is made to it. Thus saving us the trouble of doing it manually.

There are various other folders in our code. They are:

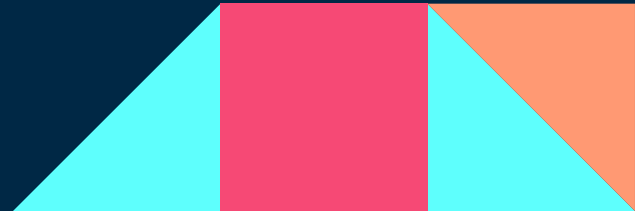
- ❑ Views: This folder contains our ejs files that serves our frontend and partials that are messages used by connect-flash to display on the screen.



- ❑ `Node_modules`: It consists of all the node packages installed by us and their dependencies, they are imported from here at various parts of the code.
- ❑ `Config.env`: This file is environment variable configuration file that consists of data like PORT number and API Key for PDFTron API.
- ❑ `Public`: The public folder consists of our static files such as css, javascript and images, etc that are required by the project for various purposes.
- ❑ `Uploads`: This folder contains uploaded as well as files generated by the API temporarily.



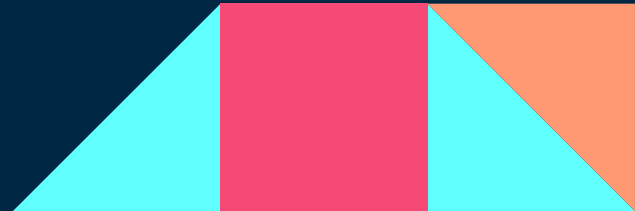
- ❏ **Package.json:** The package.json file is the heart of any Node project. It records important metadata about a project which is required before publishing to NPM, and also defines functional attributes of a project that npm uses to install dependencies, run scripts, and identify the entry point to our package.
- ❏ **App.js:** This is the backbone of our project. All the function describing how a request should be processed and responded are defined here. The functions invoking and interacting with the API are laid here and various middlewares are used in order to cater the requirements of the user accurately and efficiently.



- ❑ `Package.lock.json`: It stores an exact, versioned dependency tree rather than using starred versioning like `package.json` itself. This means you can guarantee the dependencies for other developers or prod releases, etc. It also has a mechanism to lock the tree but generally will regenerate if `package.json` changes.

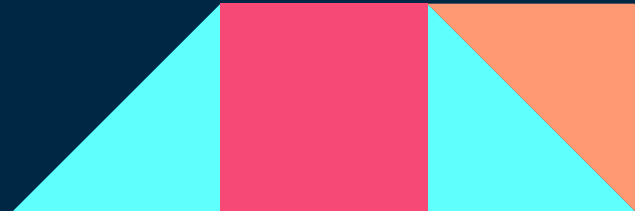
❑ PDFTron API

PDFTRON API is used to bring accurate PDF viewing, annotating, editing, creation, and generation to any web, mobile, desktop or server framework or application. We used a demo version of this api

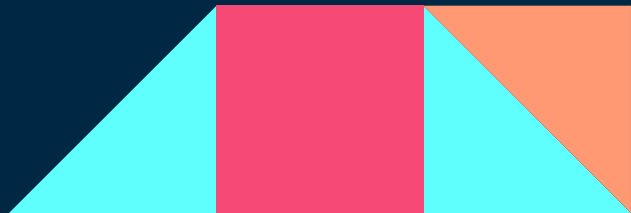


and full document and data lifecycle by deploying on our infrastructure.

We implemented it in our Node Js framework and used its features of demo/developer free version of API such as page extraction, page removal, watermark, pdf conversion. etc



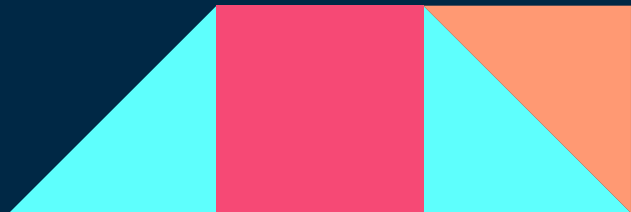
Algorithm




Interface

Our respective interface are shown in respective slides. There are in total three pages namely

- ❑ Home page
- ❑ functionality/operations Page
- ❑ Download Page



Home Page

 PDFly

Online PDF Utility

About Us

About The Project

Upload your file here

Choose file

No file chosen

Submit

Functionality/Operations Page

CONVERT

GENERATE THUMBNAIL

OPTIMIZE

Add a watermark:

Enter the text to watermark:

Submit

Extract Page No:

Page Number:

Submit

Remove Page No:

Page:

Remove

Download Page

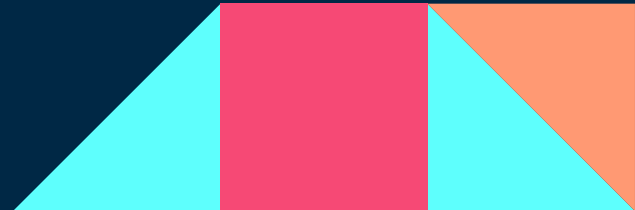
Thanks for using PDFly

You can download your file from here:

Download

References

- ❑ Node Package Manager Official website npm js: <https://www.npmjs.com/>
- ❑ Javatpoint: <https://www.javatpoint.com/expressjs-tutorial>
- ❑ PDFTron API Official Documentation:
<https://www.pdftron.com/documentation/nodejs/get-started/integration/linux>
- ❑ Stack Overflow: <https://stackoverflow.com/>



THANK YOU !!

