# BOOKPAD

**Bookpad** is a technology that provides developers with easily integratable API's for various document handling tools like

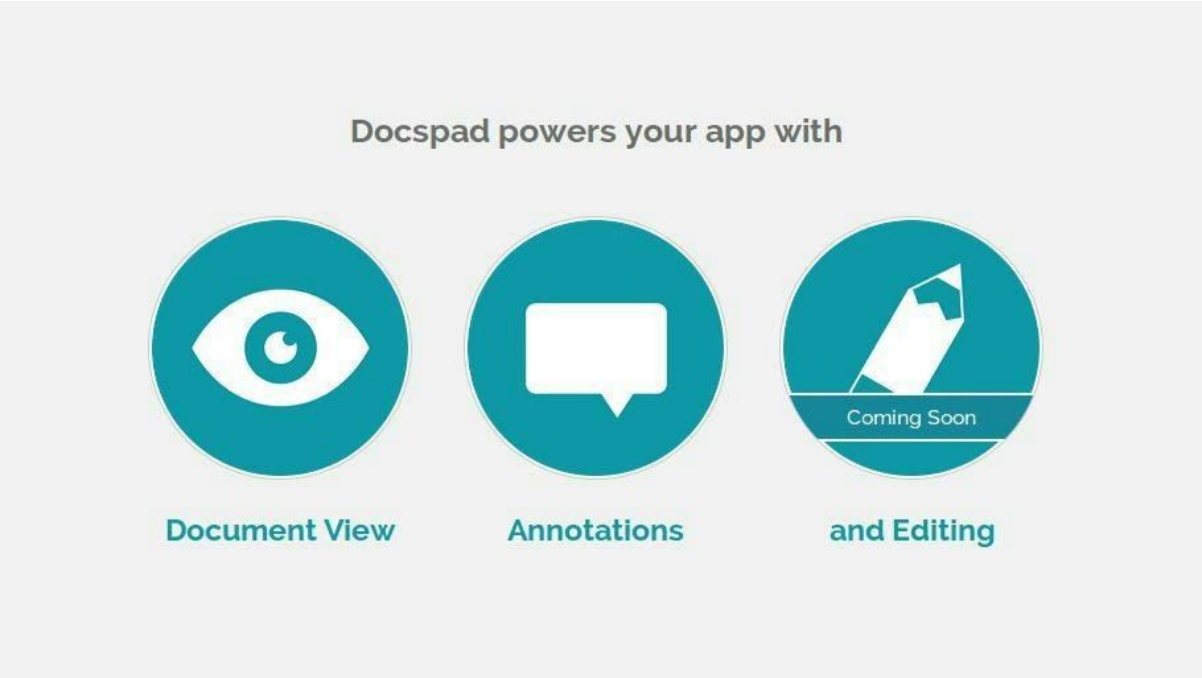
- Previewing documents
- Editing documents
- Annotating documents
- Update histories & more (which basically resides in Cloud)

A set of easy to use APIs that bring instant document handling to web, mobile and native applications

## 1.Use Cases

It can be used as a third party application to boost the services by an application.
An Example Use Case:

Dropbox allows the facility to store files but doesn't provide the facility of previewing or editing documents. So, instead of developing services from scratch, Bookpad can be used as third party service provider & the capabilities of Dropbox manifolds within couple of hours.




★ Dropbox allows the facility to store files but doesn't provide the facility of previewing or editing documents.
★ So, instead of developing services from scratch, Bookpad can be used as third party service provider & the capabilities of Dropbox manifolds within couple of hours.
★ Developers in the field of Cloud Storage, Productivity, Conferencing, E-learning and more are in urgent need of it.

**people need this for**


## 2.System

We propose to implement Bookpad technology which provides easily integratable API's for editing, viewing, annotating documents & more which basically resides in Cloud.

**Technology used**

➔ Python as programming Language.
➔ Flask, a python based micro web application framework to expose API's & request the Cloud web server.
➔ Database present in Cloud to store docs & user account details.

**Why use Flask FrameWork??**

★ Built in development server and debugger
★ Integrated unit testing support
★ RESTful request dispatching
★ Uses Jinja2 templating
★ Supports for secure cookies (client side sessions)
★ 100% WSGI 1.0 compliant
★ Unicode based
★ Extensively documented
★ Lightweight & takes care of other concurrency and similar stuffs.

**Database:**

sqlite database is used which currently acts as cloud server for the documents uploaded.

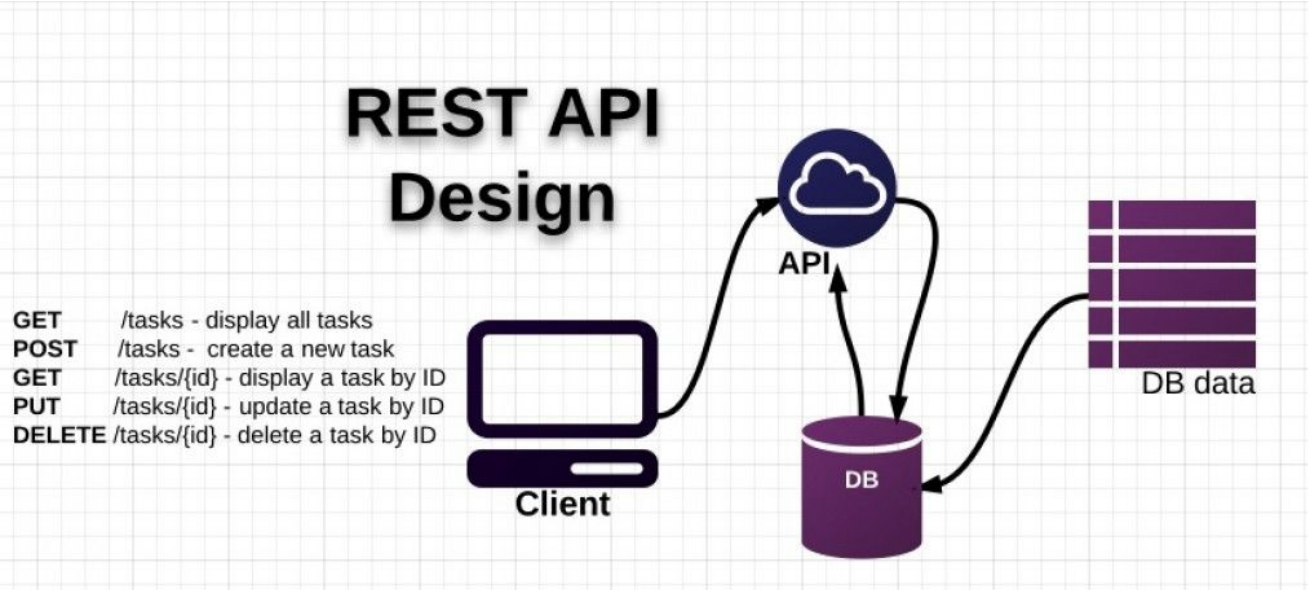
**Python modules used (for interconversion from html to pdf):**
**pdfminer:**

PDFMiner is a tool for extracting information from PDF documents. Unlike other PDF-related tools, it focuses entirely on getting and analyzing text data. PDFMiner allows one to obtain the exact location of text in a page, as well as other information such as fonts or lines. It includes a PDF converter that can transform PDF files into other text formats (such as HTML). It has an extensible PDF parser that can be used for other purposes than text analysis.

**pdfkit**

A JavaScript PDF generation library for Node and the browser.

### 3.Rest api Exposed




## API's Exposed

❖ /list **list all user's document in the cloud**

➢ URL: http://server/list/

❖ /upload **helps to upload document to the cloud**

➢ URL: http://server/upload/

❖ /preview **helps to preview document in the cloud**

➢ URL: http://server/preview?id=docid

❖ /download **helps to download document from the cloud**

➢ URL: http://server/download?id=docid

❖ /edit **helps to edit document in the cloud itself & save the changes**

➢ URL: http://server/edit?id=docid

❖ /rename **helps to rename the document in the cloud**

➢ URL: http://server/rename?id=docid&name=newname

❖ /delete **helps to delete the unwanted documents from the cloud**

➢ URL: http://server/delete?id=docid

## 5.Conclusion and Future Work :

We added bunch of useful & handful API's for developers to perform document handling on the cloud itself.

❏ Previewing documents

❏ Editing documents

❏ Annotating documents

❏ Update histories

❏ Download documents

❏ Rename, delete documents & more In future, many more useful API's can be integrated to make the application more sophisticated and useful.