

# RCPilots.ai Document

# 1. Technical Documentation

This app is designed for purpose to help candidates for RCPC exam based on provided data.

- **Architecture**

This app is consist of two parts, frontend and backend.

**Backend:**

**Router** all router javascript file in routes folder manage all URL requests incoming from frontend and call right controller.

**Controller** The controllers call functions of suitable services.

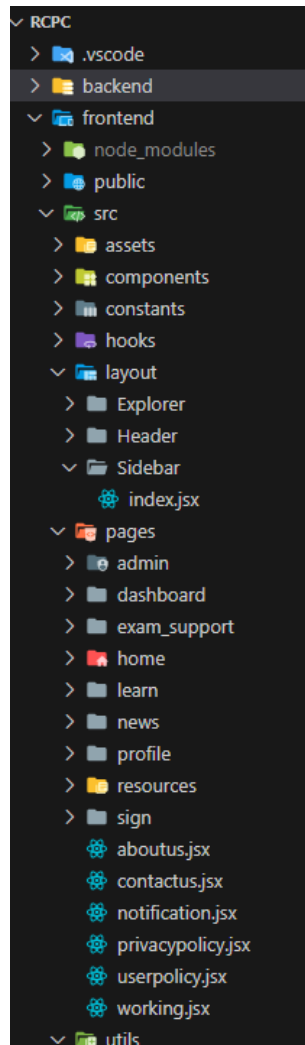
**Services & Utils :** these are the functions that execute in the end.

**Models:** this is a part related with mongoDB and in here individual database schemas are defined.

**.env:** this file includes all static informations, so it is very important

**Frontend:**

The folder instruction are public, src, .env, tailwind.config.css.



To develop the frontend so fast, tailwindCSS and Material MUI library are used.

Main code is in src folder and the code of each pages is in pages folder.

Let's look at it:

src/assets: these are resources folder like image or fonts.

src/hooks: these are react hooks

src/layouts: these are common layouts like sidebar.

src/pages: these are main user interface pages.

src/utlis: these are common utility that used frequency.

- **Components**

The languages stack is MERN and helping for candidates is realizing via ChatGPT.

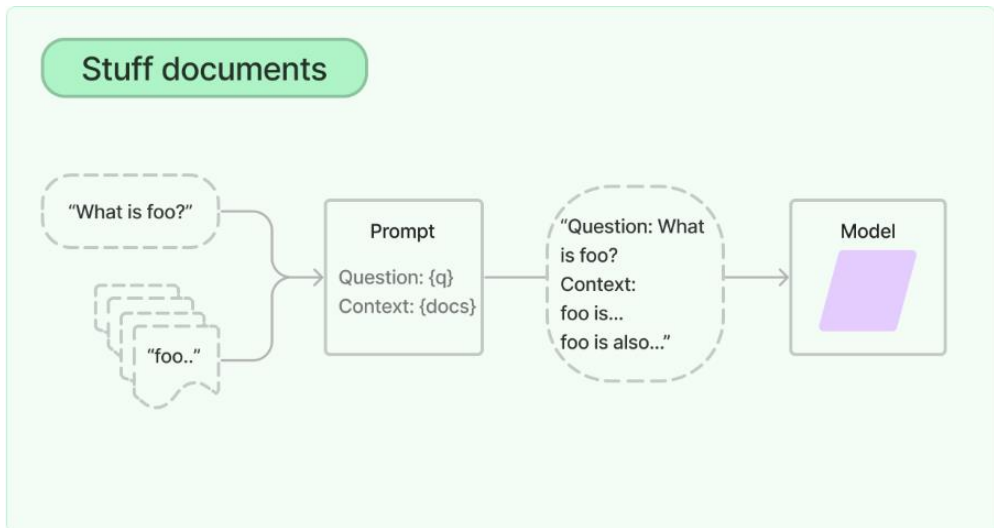
- MERN Stack: It means MongoDB + Express + React + Node.
- ChatGPT: It is an artificial intelligence (AI) chatbot that uses natural language processing to create humanlike conversational dialogue.

- **Technical specifications**

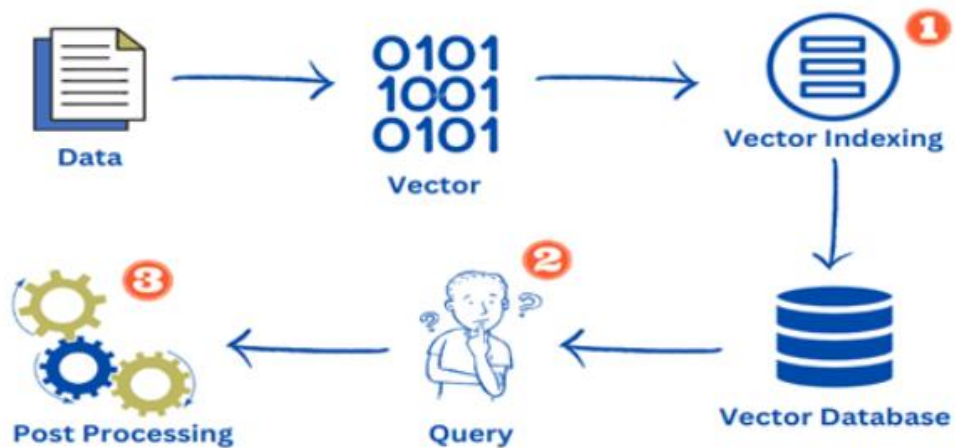
It is used langchain, Pinecone and ChatGPT.

In langchain a stuff document is used:

Pinecone is a cloud-based vector database that offers various features and benefits to the infrastructure community: Fast and fresh vector search: Pinecone provides ultra-low query latency, even with billions of items. This means that users will always get a great experience, even when searching large datasets.



Langchain stuff



Vector Database

## 2. API Documentation

- Admin : the purpose of administration

**admin/removeUser:** @param(user id)

Remove user by user's ids.

**admin/removeUsers:** @param(array of ids)

Remove users by user's ids.

**admin/ingest:** @param(http URL, doc file URL )

Scrap texts of site or doc file and save it to vector DB.

**admin/removePinecone:** @param()

Remove all pinecone DB.

**admin/removePinecone:** @param()

Remove all pinecone DB.

- Auth: the purpose of user's auth.

**auth/register:** @param(user json object)

Register user data to mongoDB.

**auth/login:** @param(user json object)

Login with user data.

- Chat: the purpose of getting answers using chatgpt and pinecone

**chat/getMainCourse:** @param(String: users question)

Generate study course with user's basic question.

**chat/getSubTitle:** @param({sub\_question:"String ...",main\_question:"String ..."})

Generate sub title with user's basic question again

- Email: the purpose of email sending

**email/sendEmail:** @param({email,message,user\_name})

Deliver email with data.

**email/contactUs:** @param({email,message,full\_name})

Send contact us message.

**email/verifyEmail**

@param({code,email})

@email:String

@code: String

@description: Verify code with the code in DB and if user is exist return user's data or not return false.

- Post: the purpose of news item

**post/addPost**

@param({content})

@content:String

@description: add posts with contents and it is displayed in news tab

**post/getPosts**

@param()

@description: Get all post and return it.

**post/getPostById**

@param(id)

Id: string

@description: Get some posts with user's id and return it.

**post/getCommentsByPostId**

@param(id)

Id: string

@description: Get some comments post's id and return it.

**post/comment**

@param({postId, content})

postId: string

content:String

@description: Create comments with id of post.

- Newsletter : create Newsletter email and send it to users

newsletter/updateSubscribe

@param({content})

@content:String

@description: Update user's Newsletter Subscribe feature

newsletter/getSubscribe

@param()

@description: Get subscribe status

newsletter/getAllInfo

@param(id)

Id: string

@description: Get all subscribed user's info

newsletter/getTemplatesinfo

@param(id)

Id: string

@description: Get individual template info

newsletter/updateStatus

@param({postId, content})

postId: string

content:String

@description: Update subscribe status

newsletter/createTemplate

@param({postId, content})

postId: string

content:String

@description: Create new Newsletter template

newsletter/removeTemplate

@param({postId, content})

postId: string

content:String

@description: Remove one Newsletter template

newsletter/updateSelected



@param({postId, content})  
postId: string  
content:String  
@description: Update selected Template

newsletter/urltest  
@param({postId, content})  
postId: string  
content:String  
@description: Test API endpoint

newsletter/updateOldeUser  
@param({postId, content})  
postId: string  
content:String  
@description: Update Subscribe status of old users

- Profile: the purpose of CURD of user profile.

profile/update  
@param({user's object})  
@description: Update user's profile with json object

profile/changePassword  
@param({new\_password, old\_password})  
new\_password:String  
old\_password:String  
@description: Update user's password with new password.

profile/uploadProfilePDF  
@param(file)  
File:PDF file  
@description: Get information from PDF and return data.

profile/updateAvatar/new  
@param(file)  
File:Image file  
@description: Create user's avatar with image

profile/updateAvatar/edit

@param(file)

File:Image file

@description: Update user's avatar with image

profile/get

@param()

@description: Get all profile information and return it.

profile/isExistEmail

@param(email)

Email: String

@description: Check if email is exist in DB and return true/false.

profile/generateCodeForPwd

@param(email)

Email: String

@description: Generate verification code when user forgot the password.

profile/resetPassword

@param({email,password})

Email: String

Password:String

@description: Reset user's password matched email.

### 3. All account that we use

- Pinecone  
PINECONE\_API\_KEY=  
PINECONE\_ENVIRONMENT=  
PINECONE\_INDEX\_NAME=  
PINECONE\_NAME\_SPACE=
- Mailgun  
MAILGUN\_API\_KEY =

MAILGUN\_DOMAIN =  
Domain login password

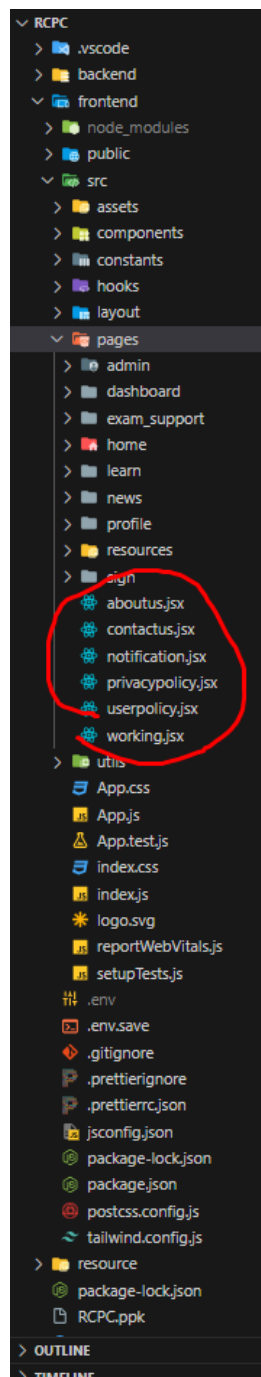
- Open AI

OPENAI\_KEY1=

OPENAI\_KEY=

OPENAI\_API\_KEY=

## 4. Instruction on how to make changes to the content



Go to the frontend folder of project resource and update the content of jsx file as above image.

For example, to change the text on the policy page, open it with notepad or code editor and update the text.

When you need to update contents, you need to clone Github repo and update contents and commit it on Github.

Then restart server.