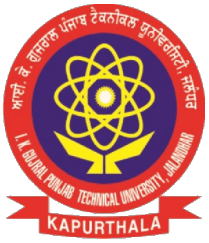
TRAINING REPORT
OF
ONGOING SEMESTER TRAINING

Submitted to



I.K. GUJRAL PUNJAB TECHNICAL UNIVERSITY

KAPURTHALA

In partial fulfilment of the requirement for the
award of degree of
Bachelor of Computer Science and Engineering (CSE)

Under the Guidance of:

Name : Mr. Shashank Agarwal
Designation : CEO
MagicAPI Inc.

Submitted by:

Name : Tarun Kumar
Roll No : 2008532

Department of Computer science and Engineering
Swami Vivekanand Institute of Engineering and Technology
Ramnagar, Banur

Batch : 2020-2024

TABLE OF CONTENTS

DECLARATION

I **"Tarun Kumar (2008532)"**, hereby declare that the dissertation entitled Training Project Report submitted for the B.Tech Degree is my original work, and the dissertation has not formed the basis for the award of any degree, associate ship, fellowship, or any other similar titles.

SIGNATURE:

Tarun Kumar (2008532)

ACKNOWLEDGEMENT

Nothing solid can be accomplished without the right mix of inspiration and planning. There is no way to complete any task without the assistance of specialities. Only insightful intellectual critiques can aid in the transformation of a project into a high quality output.

We are grateful to several persons who contributed directly or indirectly to the production of this work and who inspired my thoughts, conduct, and actions during my study.
I express my gratitude to Mr ***Shashank Agarwal,*** CEO for their support, cooperation, and motivation provided to us during the training for constant inspiration, presence, and blessings.

Lastly, I would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improve our quality of work.

Name : Tarun Kumar

Dated : 4th May 2024

PREFACE

A student acquires theoretical information in the classroom and practical knowledge on the job.When these two aspects of academic knowledge and skills are combined, a student is fully prepared to achieve his or her full potential.

There is a combination of practical experience.

When students complete a project study in an industry, they are exposed to and gain experience from real-world workplace situations. The purpose of Industrial Training is to give students the opportunity to learn about the practical side of technology in a company setting. It allows you to get a sense of the organisation and its mission.

MAGIC{API}

Date: 04th May 2024

TO WHOM IT MAY COCERN

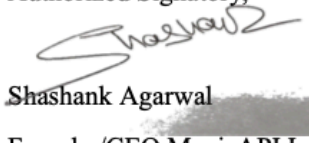
This letter is confirmation of successful ongoing remote internship by Mr. Tarun Kumar from MAGICAPI INC, San Francisco, CA. We also certify the following details:

Internship Start Date:	22 Jan 2024
Internship End Date:	Ongoing
Designation:	Software Engineer Intern

During internship, we observe him extremely inquisitive and hard working. He is very much interested to learn the functions of our core division and also willing to put his best efforts and get in to the depth of the subject to understand it better.

His association with us is very fruitful and we wish him every success in life.

Authorized Signatory,



Shashank Agarwal

Founder/CEO MagicAPI Inc

548 Market St PMB 49761

San Francisco, CA 94104

INTRODUCTION TO THE COMPANY

MagicAPI is a developer-friendly platform designed to streamline API monetization and optimize your operations. Say goodbye to the complexities and challenges of building your own platform. With MagicAPI, you can start monetizing your APIs in minutes!

MagicAPI offers a suite of features, including authentication of API calls, customer usage tracking, API log ownership, and direct customer support, all of which help developers to better understand and optimize their services. Additionally, MagicAPI handles billing and payments and allows developers to offer customized pricing plans to their customers. With MagicAPI, developers can focus on delivering a high-quality API while enjoying the benefits of streamlined monetization.

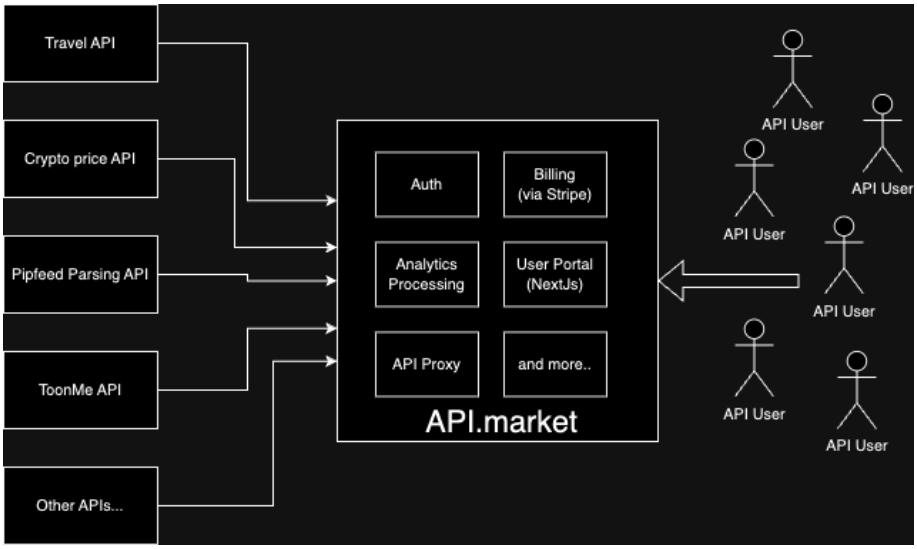
With MagicAPI, you're not just another API Provider – you're the master of your API universe.

- **Authentication:** Enable Your Customers To Generate API Keys
- **Billing:** Manage All Your Invoices, Payments, Payouts, Taxes and more.
- **Admin Portal:** Manage your API store, products, users, and more.
- **Admin Analytics:** Keep Up With The Total API Usage Or Monitor Each Client's API Usage
- **User Portal:** Let your users self onboard themselves with an easy-to-use fully customizable User Portal.
- **Direct Customer Engagement:** Build strong relationships and deliver unparalleled customer support.
- **Real-time Insights:** Gain valuable analytics and usage data to optimize your offerings.
- **Support Center:** Allow Customers To Contact You When An Issue Arises
- **Custom Branding:** Implement Your Logo, Brand Name, Contact Details, and email To personalize your API Store.

PROJECT ASSIGNED

What is API.market?

API.market is a platform where you can buy or sell Application Programming Interfaces (APIs) online. It's great for developers, companies, and individuals who want to share or make money from their APIs. API.market aims to help the API community grow by offering free listings for any API. Most listings include a free trial version so users can try before they buy. The interface is easy to navigate on API.market, allowing you to compare APIs based on price, popularity, ratings, and categories. By bringing buyers and sellers together in this marketplace, API.market makes it simpler to find and integrate new APIs into projects, making it an important resource for developers and businesses.



Working of API.market

What is an API?

API full form is an Application Programming Interface that is a collection of communication protocols and subroutines used by various programs to communicate between them. A programmer can make use of various API tools to make their program easier and simpler. Also, an API facilitates programmers with an efficient way to develop their software programs. Thus api meaning is when an API helps two programs or applications to communicate with each other by providing them with the necessary tools and functions. It takes the request from the user and sends it to the service provider and then again sends the result generated from the service provider to the desired user.

A developer extensively uses APIs in his software to implement various features by using an API call without writing complex codes for the same. We can create an API for an operating system, database system, hardware system, JavaScript file, or similar object-oriented files. Also, an API is similar to a GUI(Graphical User Interface) with one major difference. Unlike GUIs, an application program interface helps software developers to access web tools while a GUI helps to make a program easier to understand for users.

How do APIs Work?

The working of an API can be clearly explained with a few simple steps. Think of a client-server architecture where the client sends the request via a medium to the server and receives the response through the same medium. An API acts as a communication medium between two programs or systems for functioning. The client is the user/customer (who sends the request), the medium is the application interface programming, and the server is the backend (where the request is accepted and a response is provided). Steps followed in the working of APIs –

- The client initiates the requests via the APIs URI (Uniform Resource Identifier)
- The API makes a call to the server after receiving the request
- Then the server sends the response back to the API with the information
- Finally, the API transfers the data to the client

APIs are considered safe in terms of attacks as it includes authorization credentials and an API gateway to limit access so as to minimize security threats. To provide additional security layers to the data, HTTP headers, query string parameters, or cookies are used.

If we talk about the architectures, API's architectures are:

- **REST** (Representational State Transfer)
- **SOAP** (Simple Object Access Protocol)

Both define a standard communication protocol for the exchange of messages in XML (Extensible Markup Language).

What are REST APIs?

REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways.

When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint. This information, or representation, is delivered in one of several formats via HTTP: JSON (Javascript Object Notation), HTML, XLT, Python, PHP, or plain text. JSON is the most generally popular file format to use because, despite its name, it's language-agnostic, as well as readable by both humans and machines.

Something else to keep in mind: Headers and parameters are also important in the HTTP methods of a RESTful API HTTP request, as they contain important identifier information as to the request's metadata, authorization, uniform resource identifier (URI), caching, cookies, and more. There are request headers and response headers, each with their own HTTP connection information and status codes.

In order for an API to be considered RESTful, it has to conform to these criteria:

- A client-server architecture made up of clients, servers, and resources, with requests managed through HTTP.
- Stateless client-server communication, meaning no client information is stored between get requests and each request is separate and unconnected.
- Cacheable data that streamlines client-server interactions.
- A uniform interface between components so that information is transferred in a standard form. This requires that:
    - resources requested are identifiable and separate from the representations sent to the client.

    - resources can be manipulated by the client via the representation they receive because the representation contains enough information to do so.
    - self-descriptive messages returned to the client have enough information to describe how the client should process it.
    - hypertext/hypermedia is available, meaning that after accessing a resource the client should be able to use hyperlinks to find all other currently available actions they can take.
- A layered system that organizes each type of server (those responsible for security, load-balancing, etc.) involved the retrieval of requested information into hierarchies, invisible to the client.
- Code-on-demand (optional): the ability to send executable code from the server to the client when requested, extending client functionality.

Though the REST API has these criteria to conform to, it is still considered easier to use than a prescribed protocol like SOAP (Simple Object Access Protocol), which has specific requirements like XML messaging, and built-in security and transaction compliance that make it slower and heavier.

In contrast, REST is a set of guidelines that can be implemented as needed, making REST APIs faster and more lightweight, with increased scalablity—perfect for Internet of Things (IoT) and mobile app development.

What are SaaS APIs?

SaaS APIs are the building blocks that allow software applications to communicate and share functions in a secure, scalable environment.
APIs (Application Programming Interfaces) serve as the bridge between different software programs, enabling them to interact and exchange data seamlessly.

SaaS API (Software as a Service Application Programming Interface) is a combination of technology and a business model. The API is a way of sending and receiving the data, which doesn't require a frontend app. And SaaS is a subscription-based business model for software accessible through the Internet.

PROJECT TOOLS

Technologies include in project including Python, FastAPI, Uvicorn, Docker, Kubernetes, Lens IDE, Postman, Git and Github, Gitbook, Discourse.

1. **Python** - Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. I used Python to create all my APIs in using multiple python modules.

2. **FastAPI** - FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints. I used FastAPI to develop my various API's before hosting them on API.Market

3. **Uvicorn** - Uvicorn is an ASGI web server implementation for Python. Until recently Python has lacked a minimal low-level server/application interface for async frameworks. The ASGI specification fills this gap, and means we're now able to start building a common set of tooling usable across all async frameworks.

4. **Docker** - Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers.

Docker is a tool that is used to automate the deployment of applications in lightweight containers so that applications can work efficiently in different environments in isolation. I used docker to deploy my APIs.

5. **Kubernetes** - Kubernetes, also known as K8s, is an open source system for automating deployment, scaling, and management of containerized applications. I used Kubernetes to manage the docker containers.

6. **Lens IDE** - Lens is an integrated development environment (IDE) that allows users to connect and manage multiple Kubernetes clusters on Mac, Windows and Linux platforms. I used Lens to manage organisations Kubernetes cluster.

7. **Postman -** Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster. I used Postman to test out my APIs.

8. **Git and Github -** Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development.
GitHub is a developer platform that allows developers to create, store, manage and share their code.

9. **Gitbook -** GitBook is a documentation platform that allows users to create, edit, and share knowledge bases and technical documentation using a simple and intuitive interface. I used gitbook to write organisations documentation and manage them.

10. **Discourse -** Discourse is an open source Internet forum system. Features include threading, categorization and tagging of discussions, configurable access control, live updates, expanding link previews, infinite scrolling, and real-time notifications. I used discourse to create organisations community discussion forum.

WORK DONE

Overview :

The API.Market sought to develop API marketplace with many API's developed with proper documentation for anyone to use with ease at one place. Many technologies were used to get that done including Python, FastAPI, Uvicorn, Docker, Kubernetes, Lens IDE, Postman, Git and Github, Gitbook, Discourse.

My responsibility in the project as a Software Engineer Intern was to develop certain custom API's with proper documentation available and launch it on the platform. My work included from taking APIs from idea to the end product and create clear & detailed documentation. I am also responsible to create and maintain community page of API.Market . I also worked closely with other developers to assure platform's consistency, performance and security.

Purpose:

- To develop custom APIs to provide product to multiple customers that needs certain use cases API's.
- To provide every user with detailed and well written documentation.
- To provide a space for everyone to discuss and keep up with new announcement through organisations community portal.
- To give clients more user-friendly experience while trying out multiple API's.
- To make the entry barrier of the user low by removing the tedious and difficult steps it takes to test our products.

PRODUCT

There are multiple API's that API.Market host and it keeps growing everyday but here are some that I worked and developed.

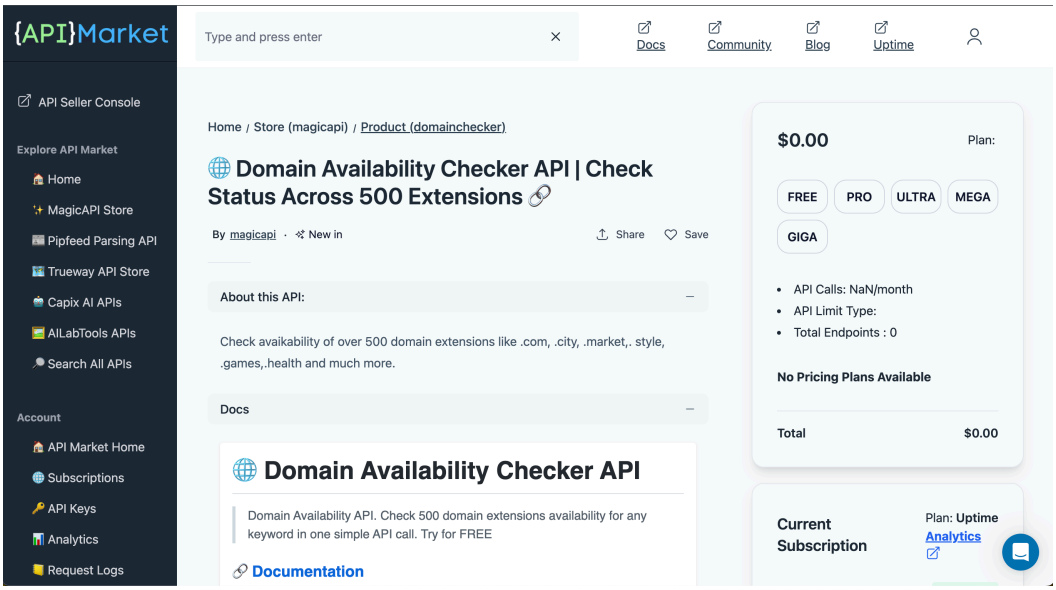
1. **Domain Availability Checker API :**

This API seeks to ease the trouble it takes to check whether a domain extension is taken or not by automating it and checking 500 domain extensions to see if a domain name is "available" or "taken". This is a quick way to check without going to each and everyone of them.

The output format of the API response is a JSON object, categorizing domain names as either 'available' or 'taken'. This structured format allows for easy parsing and integration with your applications.

Use cases for our Domain Availability API include domain registration platforms, brand protection services, and market research tools. By automating the domain availability check process, our API saves valuable time and resources, enabling you to make informed decisions quickly.

Whether you're building a domain search tool, enhancing your brand's online presence, or conducting market analysis, our API provides the functionality you need to succeed. Streamline your domain availability checks and stay ahead of the competition with our Domain Availability API.

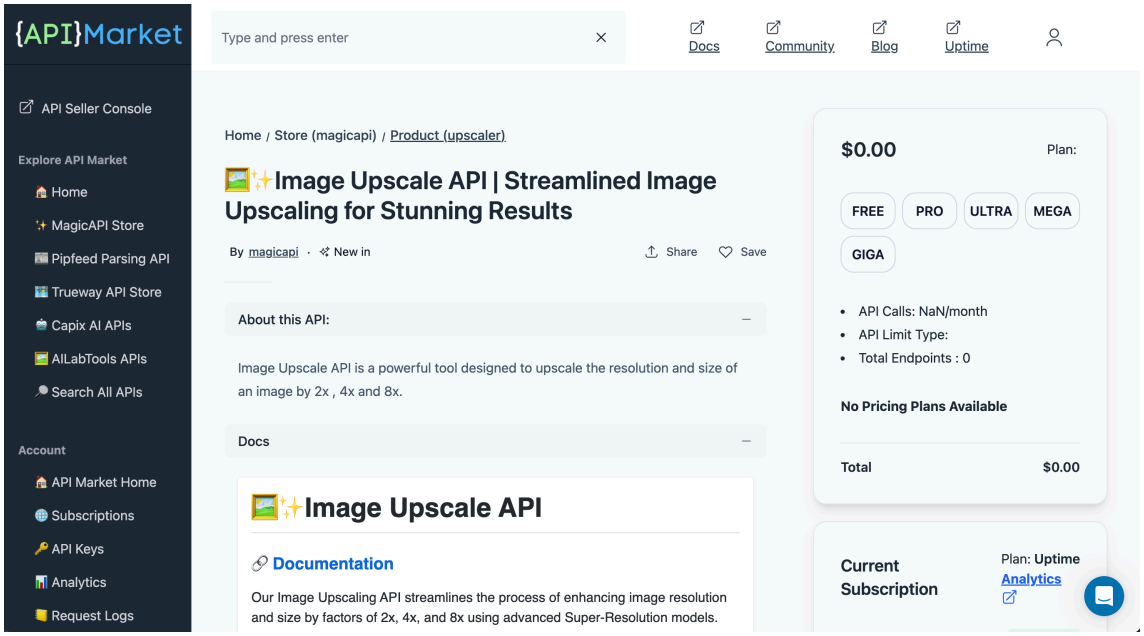


2. **Image Upscale API :**

This API seeks to remove the frustration of smaller image that gets distorted when enlarged. This API takes the smaller images as an input and then upscale them in size by 2x , 4x or 8x by just simply calling an endpoint for either one of them.

This API uses Super Resolution model to upscale the image and return the upscaled image in a JPEG format. This automates the process of upscaling the image from small non-useful images to large images.

Integrating our API into your workflow is seamless. Simply send a POST request to the respective endpoint—/upscale2x/ , /upscale4x/, and /upscale8x/ —with the URL of the image to be upscaled in the request body. The API then applies the chosen upscaling factor to enhance the image resolution.



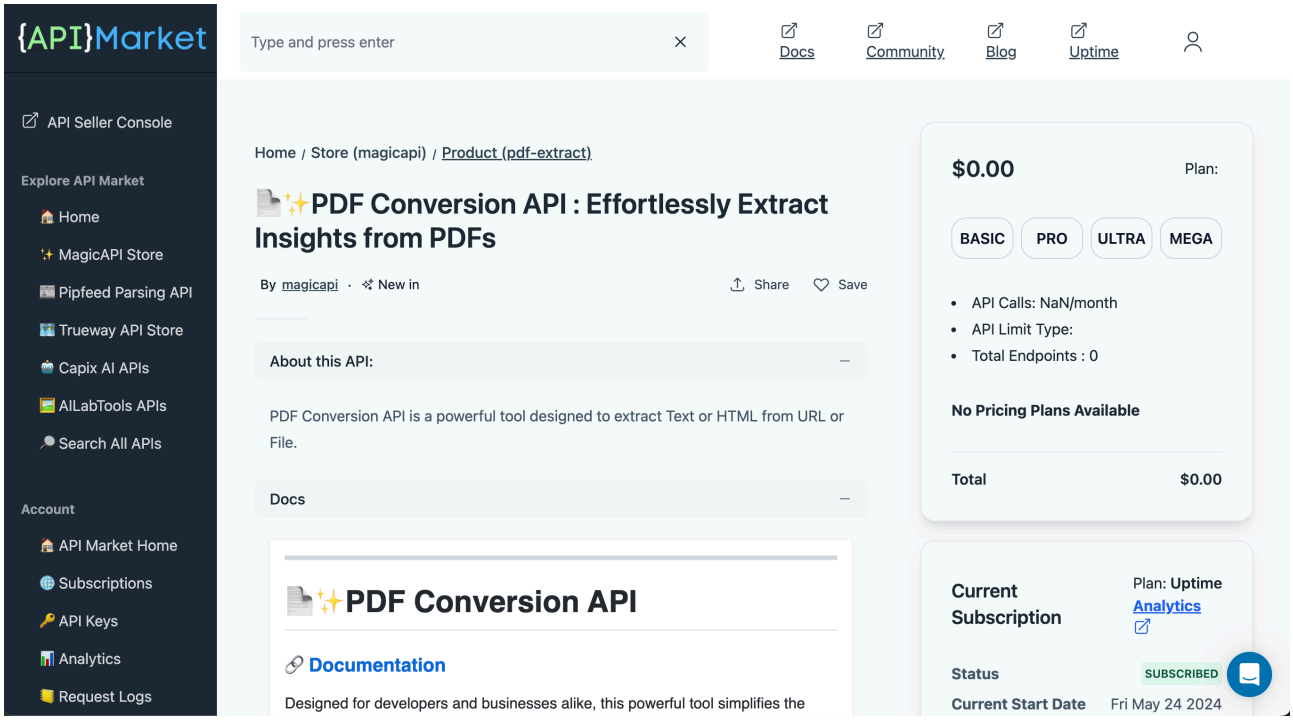
3. **PDF Conversion API :**

This API simplifies the effort it takes to extract the text from any PDF. This API extracts the text from the provided PDF either in text format or in plain HTML.

With its 4 endpoints it extracts the text or HTML of the PDF. You can upload the pdf and then provide the pdf to this API or you can simply pass the URL of the location where the PDF is located and this API does its job.

The API returns the output in JSON format, easy to parse, understand and integrate it to your application.

The 4 endpoints of the API are /pdf-to-html-url/ , /pdf-to-html-file/ , /pdf-to-text-url/ , /pdf-to-text-file/ . Using our API is a breeze. Simply integrate it into your application or workflow, and you're ready to go. Whether you're working with URLs or file uploads, our endpoints make it effortless to convert PDFs on the fly. Need to extract text from a PDF hosted online? Use the '/pdf-to-text-url/' endpoint. Want to convert a PDF file stored locally? The '/pdf-to-text-file/' endpoint has you covered. Similarly, if you're looking to generate HTML from PDFs, our '/pdf-to-html-url/' and '/pdf-to-html-file/' endpoints are at your service.



CONCLUSION

Conclusion :

The goal of this project was to provide API's with proper documentation and a place for people to come looking for announcements and discuss future prospects. As a Software Engineer Intern my role was to help develop certain custom API's for API.Market and host them on platform using technologies like Docker, Kubernetes. I Also created documentation of those API's and other documentation which are hosted on documentation website of the platform.

The project was completed for those API's and more are under work. The project was completed to present the pleasing experience for the end users and make it easier for them to use the API Marketplace.

Future Scope :

- This project with these API's have laid the foundation for multiple more API's to be developed and launched on the platform.
- Improving latency and performance of the platform and developed APIs through code optimisation and better implementation of certain items.
- Improving the readability of the Documentation to be more informative and understanding to improve the experience of the end user.
- Developing the community page into a bigger discussion forum with many more topics.
- Providing more options for the end user in terms of API's to choose from and use.
- Improving the security of the platform and keep improving the user experience.

REFERENCES

1. Python documentation : https://www.python.org/doc/
2. FastAPI documentation : https://fastapi.tiangolo.com/tutorial/
3. Uvicorn documentation : https://www.uvicorn.org/deployment/
4. Docker documentation : https://docs.docker.com/guides/
5. Kubernetes documentation : https://kubernetes.io/docs/home/
6. Lens IDE documentation : https://docs.k8slens.dev/
7. Postman documentation : https://learning.postman.com/docs/introduction/overview/
8. Git documentation : https://git-scm.com/doc
9. Github documentation : https://docs.github.com/en
10. Gitbook documentation : https://docs.gitbook.com/
11. Discourse documentation : https://meta.discourse.org/c/documentation/10
12. API. Market : https://api.market/
13. API.Market documentation : https://docs.api.market/
14. API.Market community page : https://community.api.market/