

Contents :

1.Use Case.....	03
2.Technology Used.....	04
3.Requirements.....	06
4.Solution Brief.....	06
5.Work Flow.....	07
6.Architecture.....	08
7.Graphical User Interface(GUI).....	09
8.Testing Credentials.....	11

Use Case :

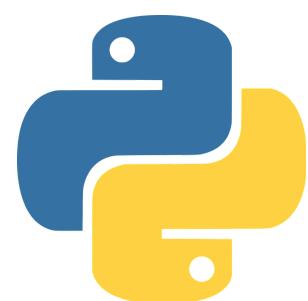
Enterprise AI :

With Superior Customer Experience of paramount importance for all business operations, it has become imperative to treat this process with utmost care infused with latest AI/ML technologies. It is expected to drive the customer experience to a new level of awareness and acceptance. The sea of technologies changes, have also brought to the table, newer and better aspects towards our customer experience. These customers include the internal as well as external customers.

- Create UI that would allow you to provide a Structured (Databased) or and Unstructured (Audio /Video /Text /Image) input, as the case may be, to your AI solution.
- Create the AI algorithm that would address your chosen use case and draw insights from the content.
- Create a backend database or a content store if required for your use case.
- Create an Outcome UI that would showcase the results from your AI algorithms. Data visualization/analytics can also be shown in the UI (like charts, graph, tables etc...)

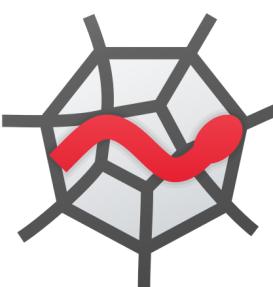
Technology Used :

Backend:

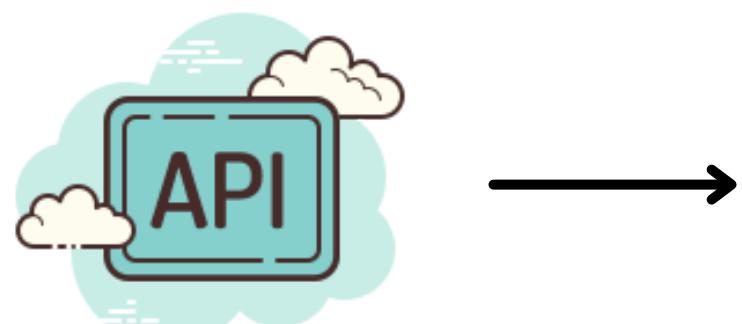


Python:

Used Python3 for backend of the web application.



Used Anaconda's Spyder (IDE) for Coding.



Used FAST2SMS RESTful API Services for SMS notifications.



Used Apache for Serving Web Application.

Framework:



Flask:

Used Python Flask for framework of web application.

Database:



Used freemysqlhosting MySQL database (5MB) services for database of web application.

Technology Used :

Frontend:



Used HTML & CSS for Web Application Structure & UI Design.
Also used third party libraries like Bootstrap and Fontawesome for a better design.
And used JS to configure google pie chart for stock analysis.

Frontend Framework:



Used Jinja2 as medium between Python and frontend of web application.

PaaS:



Used Heroku Cloud as Platform-as-a-Service (PaaS) to run web application on online server.
(Project uses 52.2MB of Cloud Storage)

Requirements :



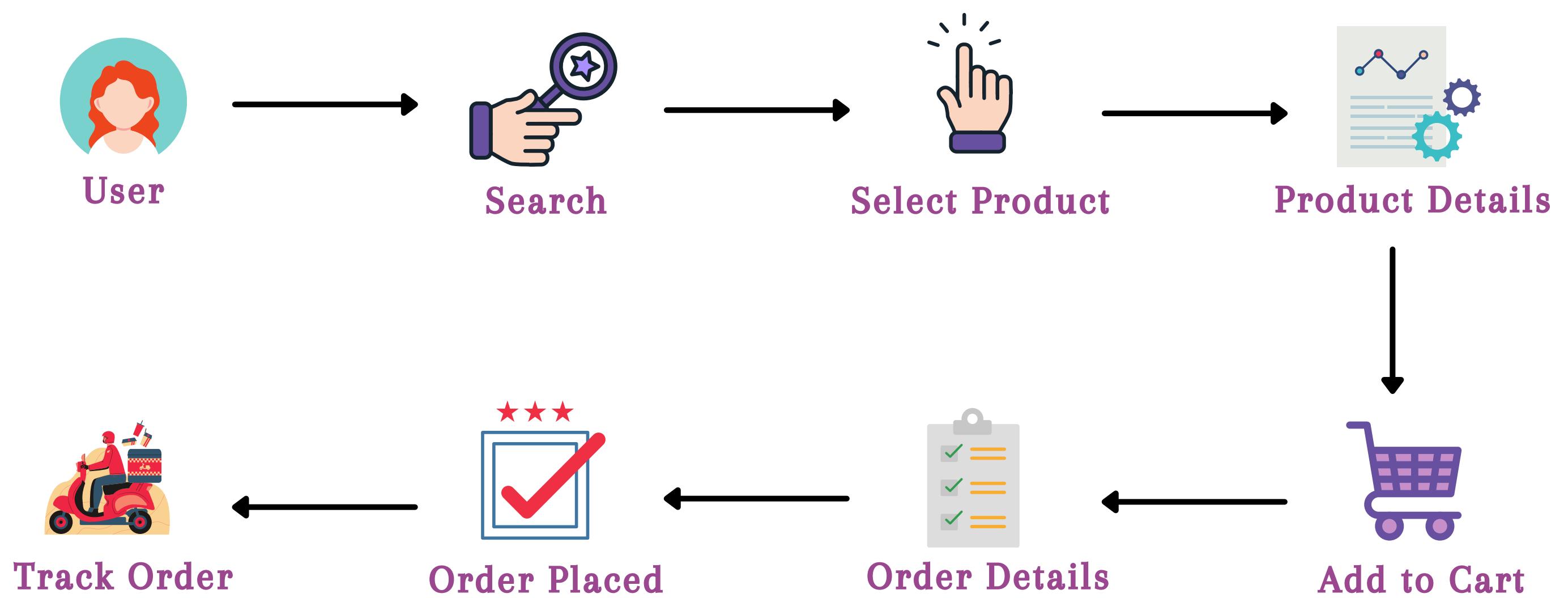
Requirements to view or access this web application are just a web browser with HTML5 support and a stable internet connection.

Solution Brief :

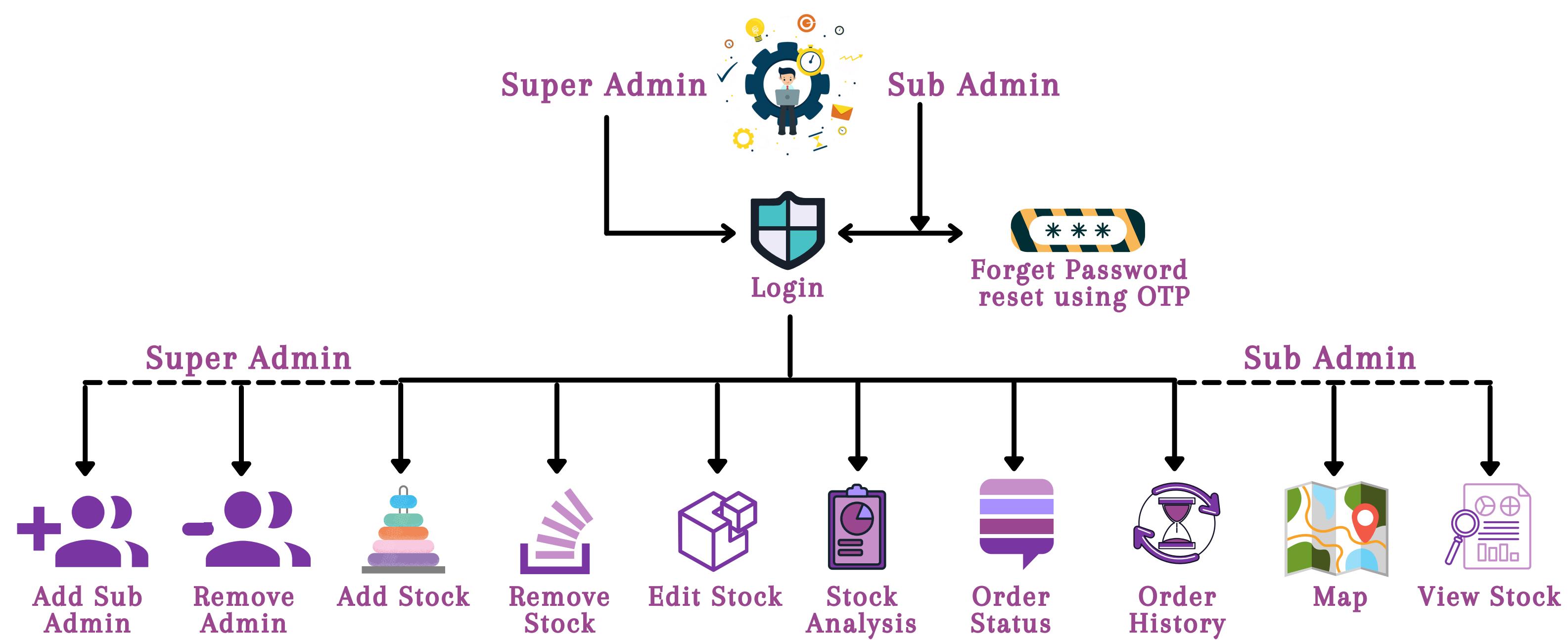
The "Book Store" is a complete ecommerce solution where a user do operations like search, categorized search, add to cart, place order, track order with a very user friendly User Interface (UI). Web application has a superb search algorithm based on string. Web application also have admin logins to maintain and analyze the stock, orders and set product recommendations. Web application have a concept of SuperAdmin and SubAdmin where a SuperAdmin have control to add or delete the SubAdmins. The web application is equipped with automated SMS and email notifications system. Web application also generates OTP for various tasks like forget password.

Work Flow :

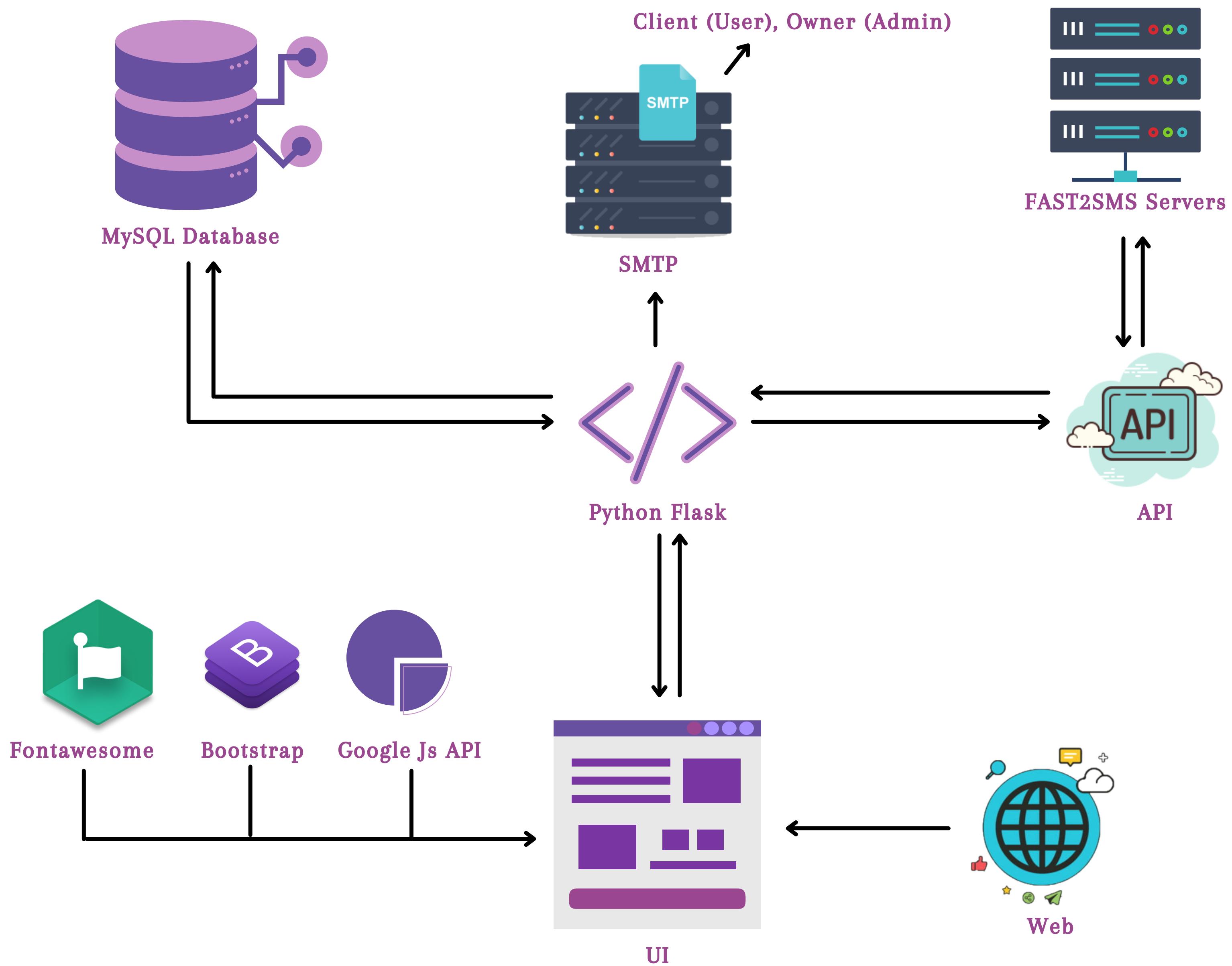
User side work flow :



Admin side work flow :



Architecture :



Used database for managing data of stocks, users and orders.
SMTP used to send autogenerated email notifications when - Order placed, Contact Us,(Each time mail send to owner (admin) and user both) and OTP mail for forget password.
API - Using RESTful of API application sends automated Order placed SMS to buyer.
All this managed by Python FLASK framework.