Full Stack Project :-

→ Imm Immplementation of Reccomendation System :-

o Lets understand with an example:-

We have 4 sentences:-

→ I love Troll2!
I like Troll2.
I love Troll2!, I love it!!! ⟶ **Similar**

Troll2 is bad.

o Incase of longer data we need computer to find Similarity :-

↳ Cosine Similarity :- is reletively easy to calculate metric that can tell us how similar or different vectors are.
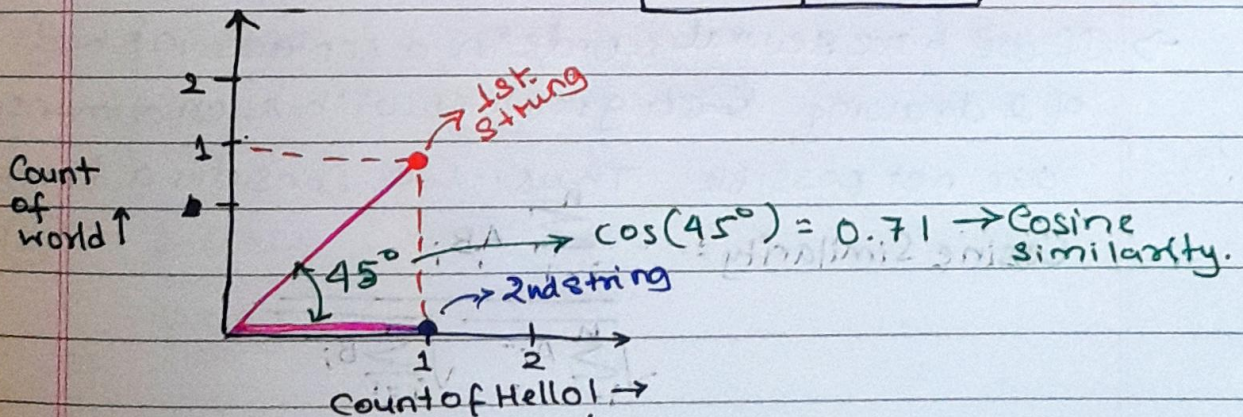
Lets take an example:-

① Helloo world!
② Hello!

	Hello	World
①	1	1
②	●1	0

← Hello world!
← Hello 0.

$\cos(45°) = 0.71$ → Cosine similarity.
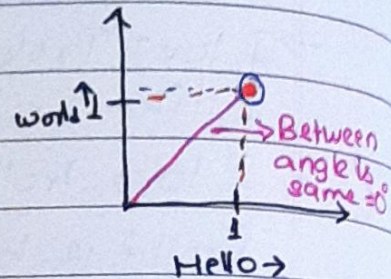
Count of world ↑

Count of Hello! →

→ If the 2nd phrase was "Hello, Hello Hello" even then only the length would change but angle remains same.

Thus :- [Cosine similarity defined by angle b/w lines not length b/w lines.]
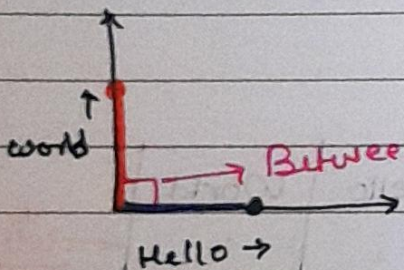
→ If both strings were "Hello world" :

	Hello	World
Hello world →	1	1
Helboe coorld →	1	1

world ↑ ... Between angle is same = 0

Hello →

[Cosine similarity ← $\cos(0°) = 1$]

→ If nothing is common →

Hello	world
1	1
0	0

world ↑

→ Between angle $90°$
$\cos(90) = 0$
∴ [→ cosine similarity = 0]

Hello →

→ If we have several words in a sentence instead of 2 drawing graph with such dimensions are not possible. Thus, we consider a formula

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \ \sqrt{\sum_{i=1}^{n} B_i^2}}$$

I	love	troll	2	Gymkata
1	1	1	1	0
1	1	0	0	1

A = I love troll2!

B = I love Gymkata!

- (A and B are different strings that we are comparing.)
- (and their values defined as A_i and B_i where $1 \le i \le n$.) $1 \to$ Presence of word, $0 \to$ Absense.

- From here we calculate:-

$$\frac{(1 \times 1) + (1 \times 1) + (1 \times 0) + (1 \times 0) + (0 \times 1)}{\sqrt{1^2 + 1^2 + 1^2 + 1^2 + 0^2} \times \sqrt{1^2 + 1^2 + 0^2 + 0^2 + 1^2}}$$

$$= 0.58 \longrightarrow \text{cosine similarity.}$$

- IN OUR DATASET :-

→ Here we are predicting 3 things → Disease, Medicine, Precautions.

→ We had 3 data sets →

①
Disease	Symp1	Symp2	.....	Symp17
Disease name	Symptoms names	....	---	...

②
Disease	Prec1	Prec2	Prec 4.
Disease name	Preection name		

③
Disease	Medicine
Disease name	Medicine name.

→ [As we need to predict disease we keep disease row names and rest as column names.]

→ The dataset Now changed :-

Disease	Prec1	Prec2 ...	Medicine •	Sym1 ...	Sym 17
Disease names	Prec names		meds names	symptom names	...

→ Now applying cosine Similarity on this matrix:-
we get same data set and its values changed
to [similarity Index]:-

if matched → ①

else → ⓞ

otherwise → cosine calculated (How much matched)

Before that as we need to predict disease
and find similarity between disease strings
We transpose the dataset formed.
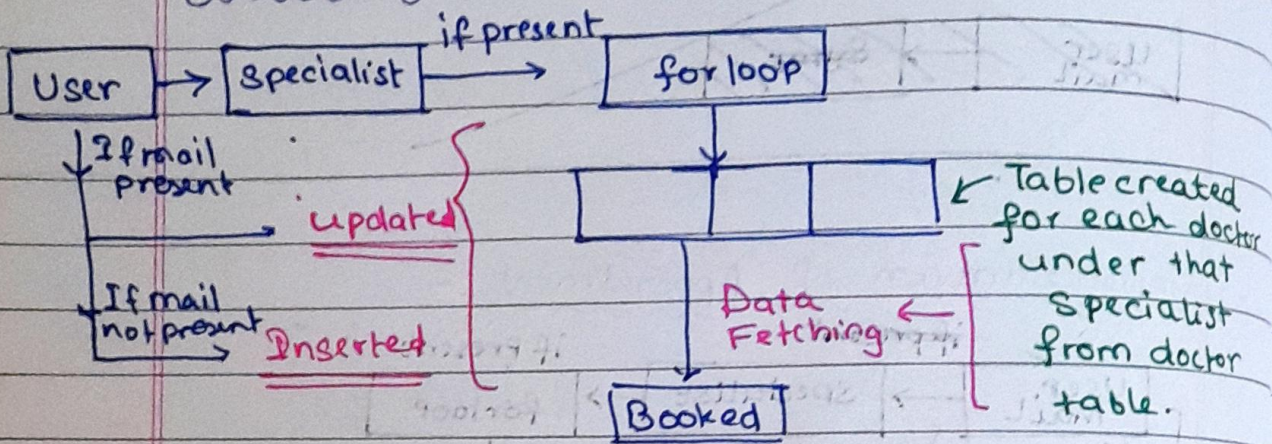
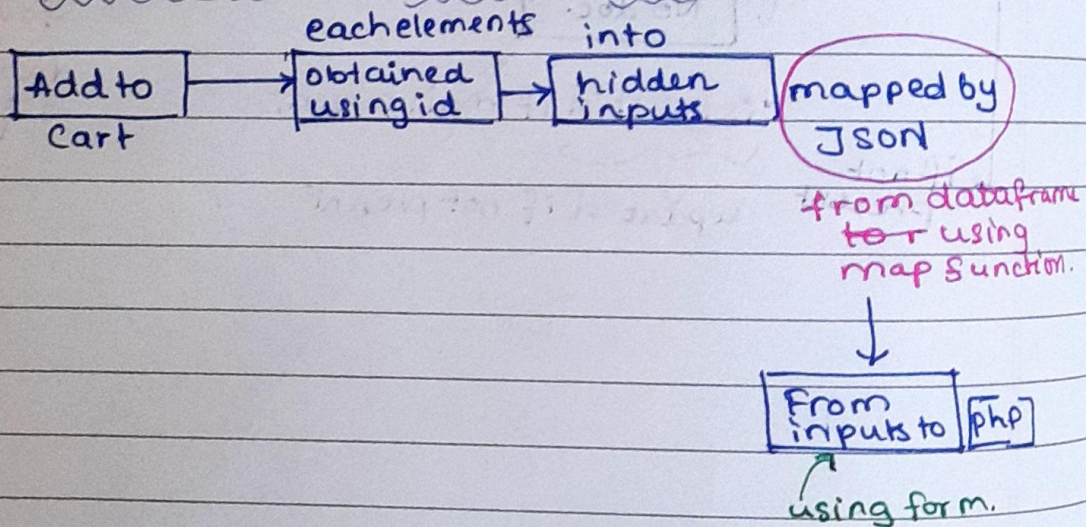| Prec	Diseases....
Sym	values ------
Med	

← This matrix gives us cosine similarity.

→ Based on this matrix and similarity obtained
we suggest most similar diseases → map it
with its respective [precautions & medicines]

Done in Tkinter app.
Later this code is
implemented in
web → using python flask

→ **Implementation of Appointment :-**

```
┌──────┐      ┌────────────┐  if present   ┌──────────┐
│ User │ ───→ │ Specialist │ ─────────────→│ for loop │
└──────┘      └────────────┘               └──────────┘
```

If mail present ────→ *updated*

If mail not present ────→ *Inserted*

Data Fetching ←

Booked

← Table created for each doctor under that Specialist from doctor table.

→ **Implementation of Json in Shop :-**

```
┌─────────┐  each elements  ┌──────────┐  into  ┌──────────┐   ⟨mapped by⟩
│ Add to  │ ───────────────→│ obtained │ ──────→│ hidden   │    ⟨  JSON  ⟩
│ Cart    │                 │ using id │        │ inputs   │
└─────────┘                 └──────────┘        └──────────┘
```

from dataframe to r using map function.

↓

```
┌──────────────┐
│ From         │  ┌─────┐
│ inputs to    │  │ php │
└──────────────┘  └─────┘
```

using form.

→ **Dashboard Backend Explanation :-**

Fetching   Google
```
┌──────────┐    ┌───────────┐
│ using    │ ──→│ chart.js  │
│ various  │    └───────────┘
└──────────┘
```
select count queries   used to plot

Data from fetch converted to (json string)

string obtained from json using (php).

Fetching

```
┌─────────┐        ┌──────────┐        ┌──┬──┬──┬──┐
│ using   │───────▶│ for loop │───────▶│  │  │  │  │  ◀── printed
│ PHP     │        └──────────┘        └──┴──┴──┴──┘      in tables
└─────────┘                                               for user details
                                                          and top doctors.
```

- JSON EXPLANATION :- (Javascript Object Notation).

  data structure
  → We use to encode(php) ~~variable~~ to json to pass it

  in js :-

  Eg: $ myObj → name = "John";
         $ myObj → city = "New York";

  $$\boxed{\$ myJSON = json\_encode\ (\$myObj);}$$

  Client Side:-
  ↳   xmlhttp = new XMLHttpRequest();
      xmlhttp.onload = function() {   displays
                                                       using JS from
      const myObj = JSON.parse(this.responseText);   PHP.
      document.getElementById("a").innerHTML =
                                           myObj.name;
      }

      ┌───┐ ◀── gets
      │ xmlhttp.open("POST", "demo_file.php");   │   the type
      └───┘   of request
      xmlhttp.send().   (POST).

  ↳   GET ◀── all details sent via form displayed in URL
            └──▶ Data security Lost.

      POST ◀── all details sent via form is hidden.
            ↳ Data security maintained.

☐ Future Scope:-

① In doctor appointment we could implement booking of doctor based on ratings and reweiw also chat for booking the doctor.

② In reccomendation ~~we can~~ ~~easily~~ dataset shd be more precise to fine tune the outcomes.

③ Based on best reccommendations → doctor can be reccomended.

④ More products and company shd be added for better shop compatibility.

⑤ Better dashboard & login.

Role of Each Members:-

1. Ritwika → Login, Sign Up, Doctor Login, Registration, Machine Learning implementation in Python using Cosine Similarity, Backend Recommend.

2. Soham → Python flask to obtain results of ML ~~imple~~ implementation to the website, Dashboard.

3. Swarnasish → Appointment of Doctor and booking of time slots. [frontend & Backend]

4. Sayantan → Medical Shop adding to cart and buying Products. [frontend & Backend]

Flow of Presentation:-

↳ Soham → [Introduction.]

↳ Ritwika → Doctor / user Login.

↳ Swomo → Home Page, contact us, About, Services.

↳ Ritwika → ML implementation Reccomendation

↳ Soham → Flask implementation, front end

↳ Ritwika → Backend Reccomendation.

↳ Soham → Prescription.

↳ Swomo → Appoint ment form

↳ Sayantan → Shop.

↳ Soham → Dashboard

↳ Soham → [Conclusion, future Scope, Role of each members.]