ATA:

Automatic_Team _Assembler

DEMO

https://ata-automatic-team-assembler.vercel.app/

Please do not close the page after finish the form, or the result will disappear!

Data Structure - Vectorization

Skill Level (Diversity)

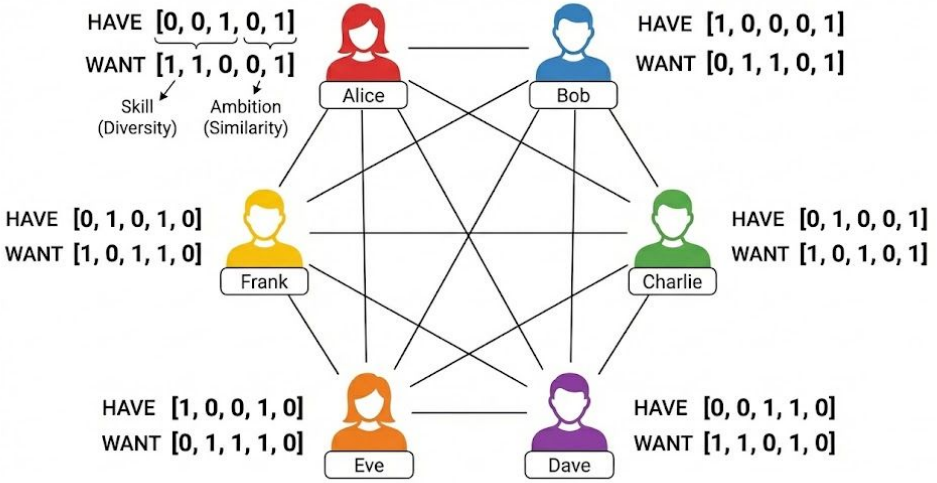
Beginner / Intermediate / Pro

HAVE: [0, 0, 1, 0, 1]

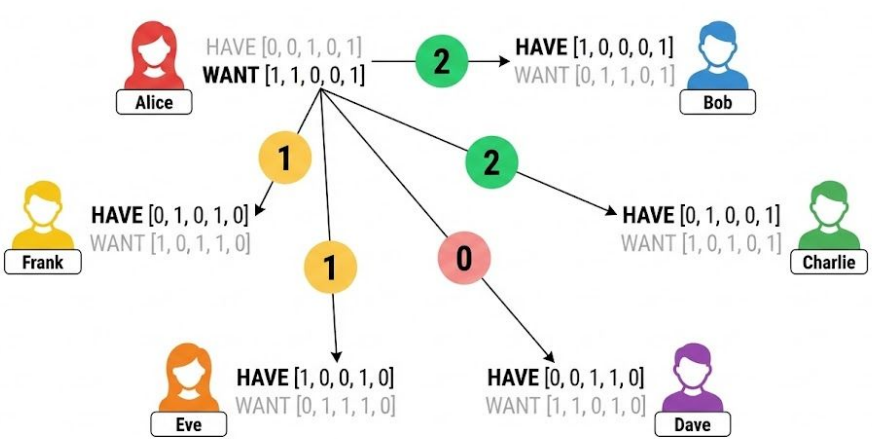
Just Pass / Pursue Excellence

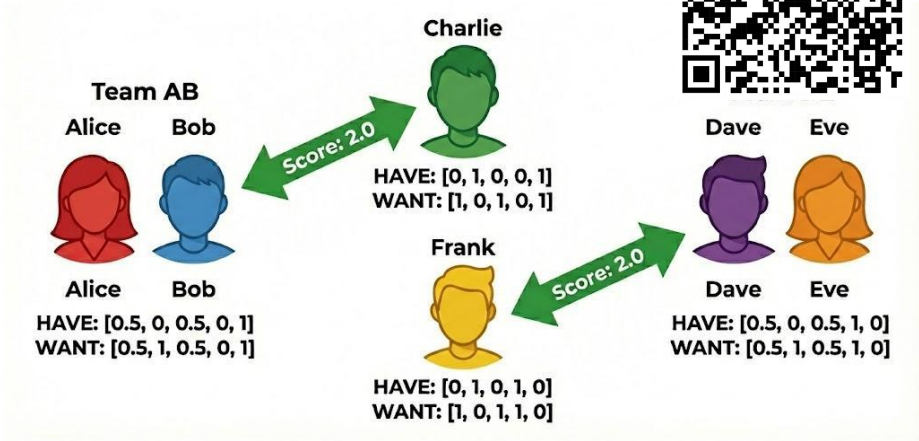
Learning Goal (Similarity)

WANT: [1, 1, 0, 0, 1]



Matrix Operations



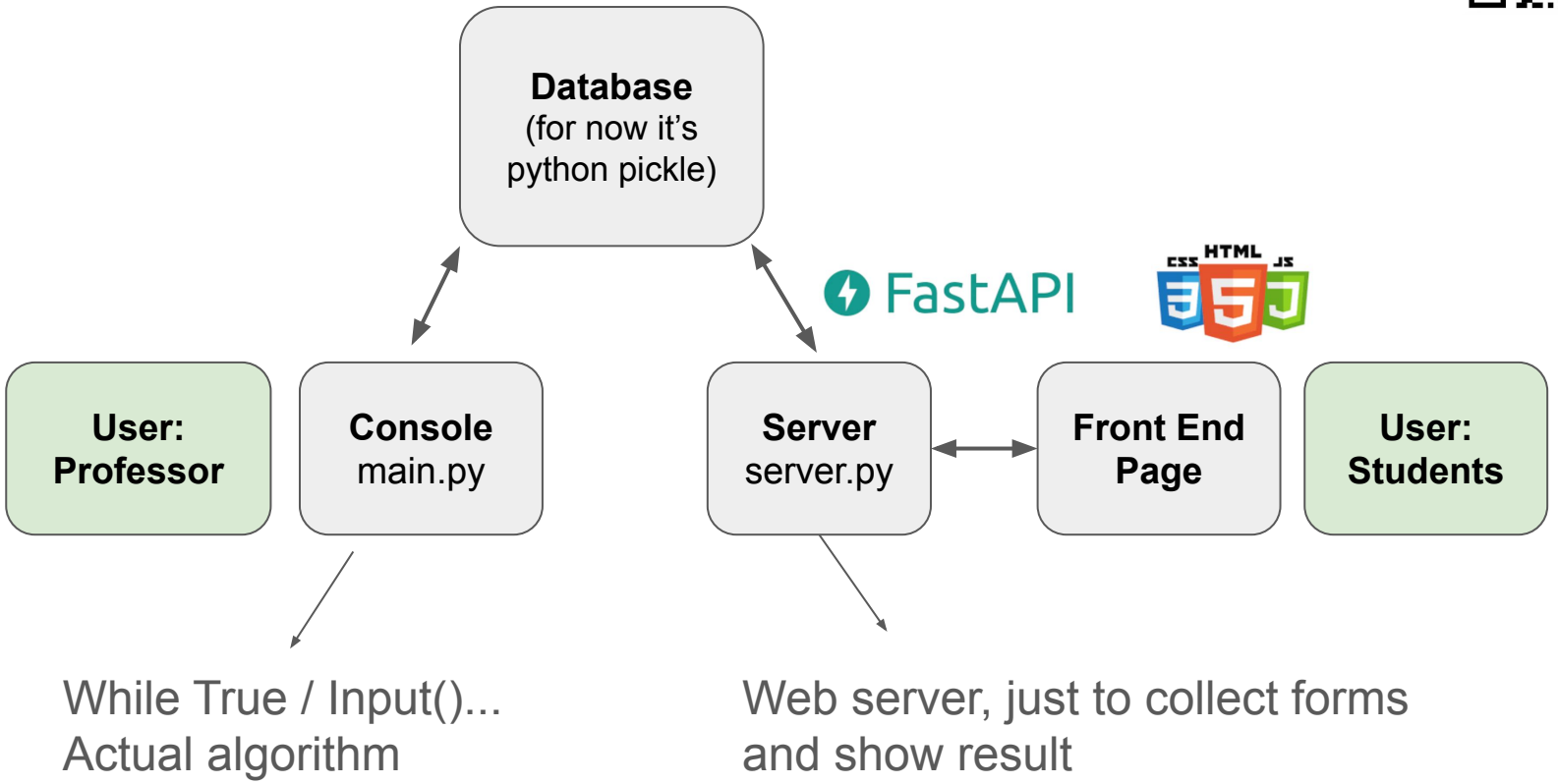


$$\mathbf{S} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} - & \mathbf{2} & \mathbf{2} & 0 & 1 & 1 \\ 2 & - & 1 & 0 & 1 & 1 \\ 2 & 1 & - & 0 & 1 & 1 \\ 0 & 0 & 0 & - & 2 & 2 \\ 1 & 1 & 1 & 2 & - & 2 \\ 1 & 1 & 1 & 2 & 2 & - \end{pmatrix} \end{array}$$

Pair score “Crush score”
= (score(AB) + score(BA)) / 2

Pair from highest to lowest:
AB = AC = DE = EF = 2
AE = AF = BC = BE = BF =CE = CF = 1
Rest = 0 scores

System Architecture & API Design



OOP Design Principle

class Student:

Properties:

- Student's property
- Student's team information

Method:

- construct_vector()

class Team:

Properties:

- Students list [Student]
- Team's vector

Methods:

- add_students([Student])
- crush_score()

class Course:

Properties:

- Students list
- Score matrix
- Team list

Method:

- team_matching()

list of Student() objects!

Student() method → *Team()* method with Student() objects → *Course()* method with Team() objects

Deployment, Persistence & Frontend

Store in docker container | Repo on Github | Deploy on AWS EC2 | Front End

Can work / develop on local, and run on remote server

Contribution and Future Work

Guanyu Tao: Algorithm / OOP architecture / Deployment
Yiying Xie: Server / Console / Data Localization

Future improvement:

- Use Postgresql for data localization
- Generalize for all other classes
- Customization for student’s profile
- Beautiful console UI