Sequence of running the project:

Service registry -> details service -> profile service -> api gateway

Registration:

{

    //"account\_no":"1222337IOB",

    "firstname":"yash2222",

    "lastname":"saxena222",

    "email":"saxenayash1503@gmail.com",

    "contact":7007223434

“password”:abc(autogenerate)

}

Response:

{

    "account\_no":"1222337IOB", //auto generate

    "firstname":"yash2222",

    "lastname":"saxena222",

    "email":"saxenayash1503@gmail.com",

    "contact":7007223434

}

{

"account\_no": "1222337IOB",

    "pin": "3303" //auto generate //not required

}

[to make the custom response : use modal mapper]

To add api gateway:

Create a new springboot application to the existing code(API gateway).

Just modify the pom.xml and properties file.

To check in postman:

Use the urls of APIgateway.

To enable h2 database:

Copy the dependency in pom.xml of user-details.

In properties file, add 3 properties

To make the data get reflected in the database:

#Add “modalmapper” dependency in the pom.xml (used when we want response in a specific format)

Test connection: it will come successful and try running the command

Add support to run: json and xml

Add dependency: Jackson

To enable the security:

Registration:

To encrypt the password:

Use BCryptPasswordEncoder bCryptPasswordEncoder;

Websecurity

Login:

To understand the UI:

https://www.youtube.com/watch?v=YJ7ZgGnhN5k