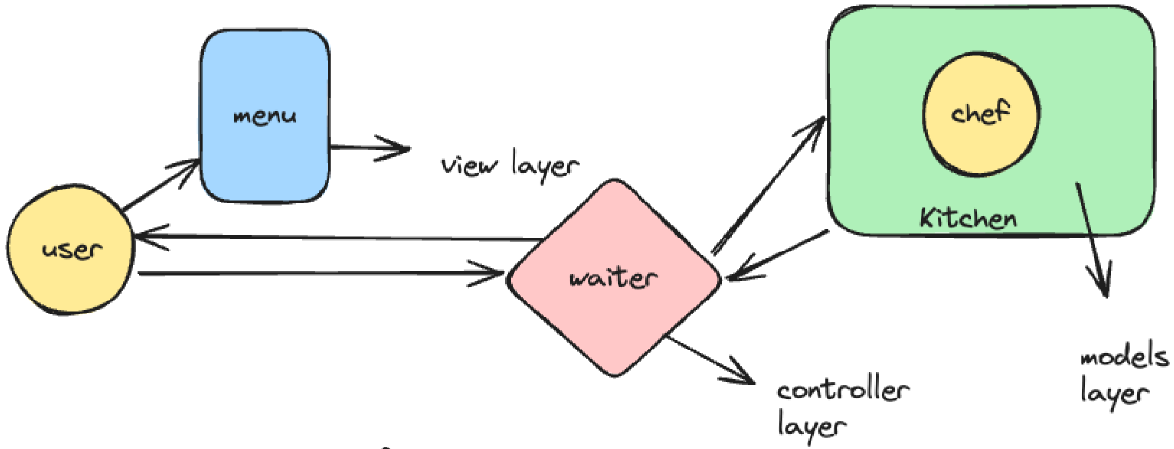
MVC

Backend Architectures : MVC

Restaurant ----> You enter as a user



Separation of concern

A similar technique can be used to prepare backend code bases.

MVC - Model View Controller

What do you mean by layer ?
It will be a separate set of code (may be wrapped in a directory) which will contain functions and classes for that purpose

Model Layer: Model layer contains the whole business logic of the app.

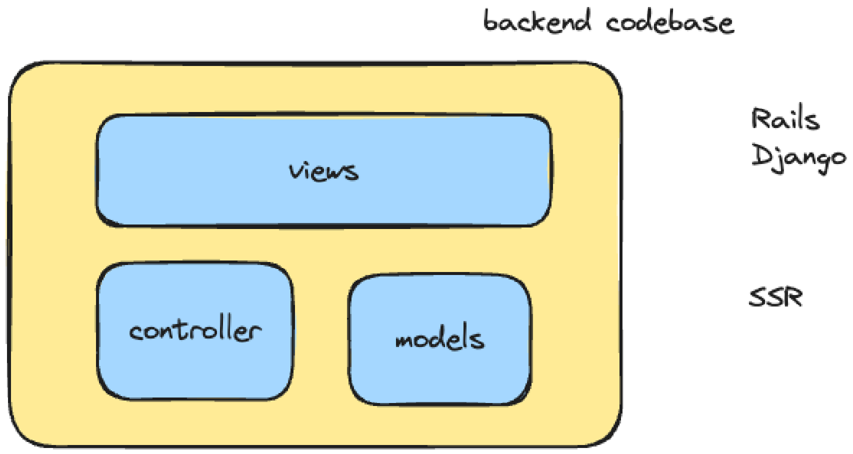
ex: 1. How to do user logic/logout
2. What should happen if two people try to book the same seat in a theatre ?
3. logic to find nearby drivers

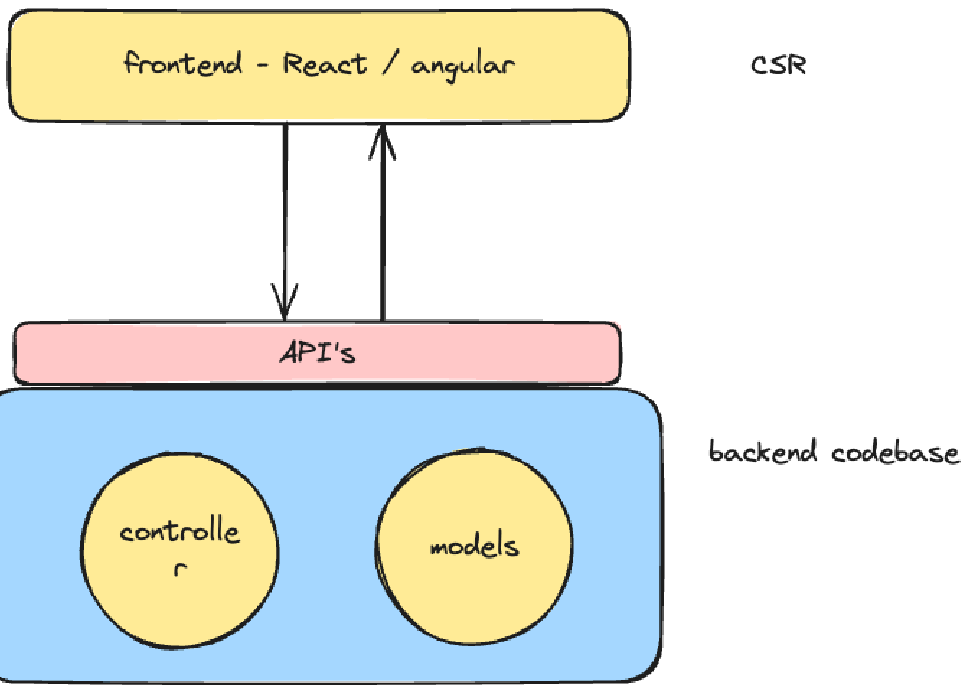
To implement business logic a lot of times, model layer needs to talk to data storage.

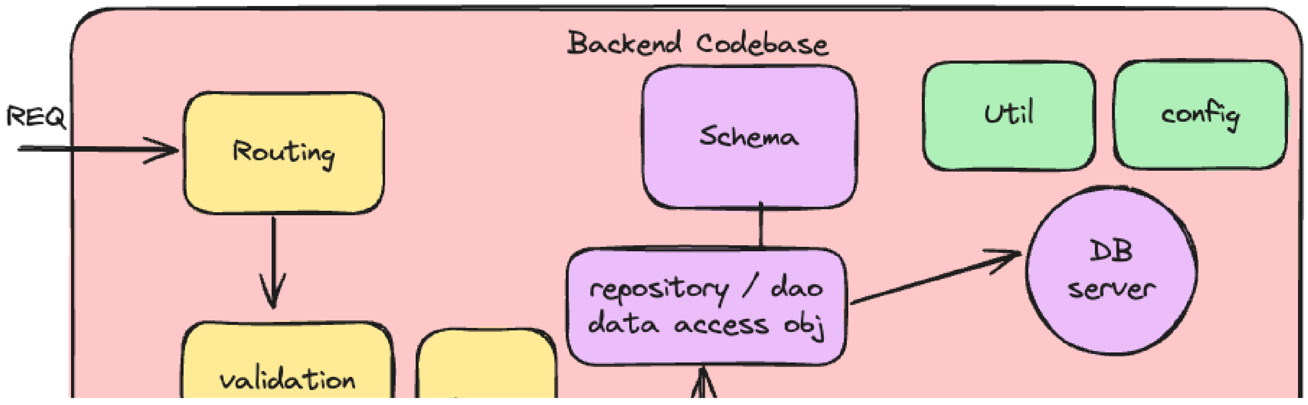
controller layer: Controller layer is responsible for accepting requests and forwarding it to the model layer. Once model layer has the result prepared, then it gives it back to controller layer which returns it to the user.

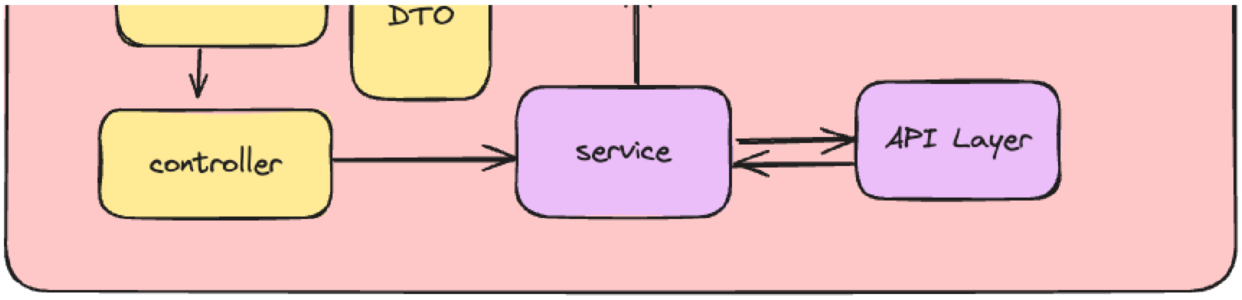
Keep your controllers thin.

view layer: Frontend









Routing layer: It contains functions and code logics that will route the incoming request to the right validators and controllers
Segregate your API driven routes with others

/api/v1/makePayment

/api/v2/makePayment

Service layer will only contain the business logic. Service layer will not directly talk to the DB layer.

respository/dao layer will be responsible for comm to the actual data storage