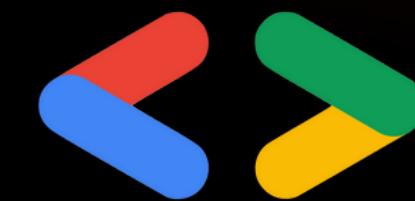


# Getting Started with Web Development

Department of Computer Science ,  
Aligarh Muslim University



**CSS AMU**



**GDSC AMU**

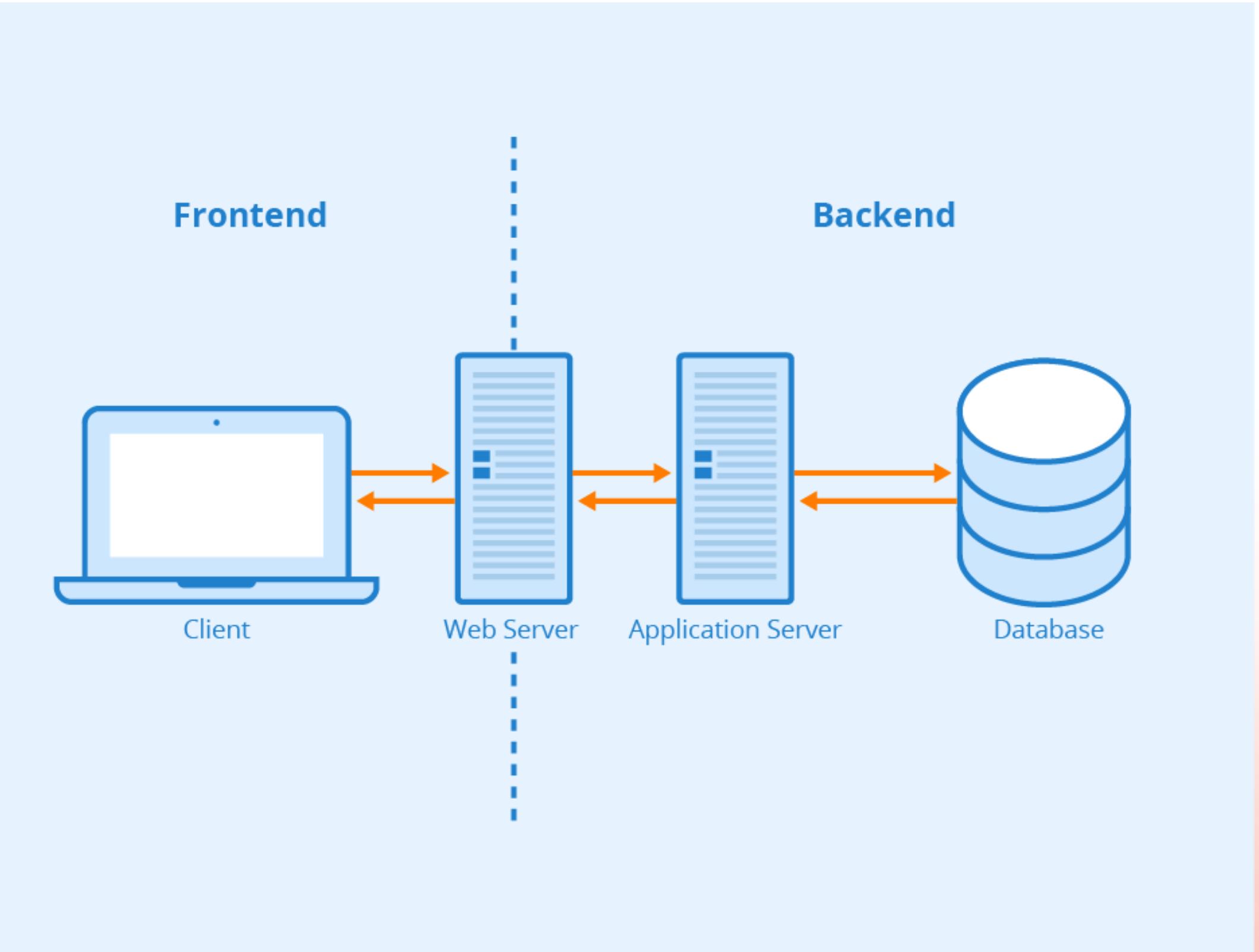
# Content

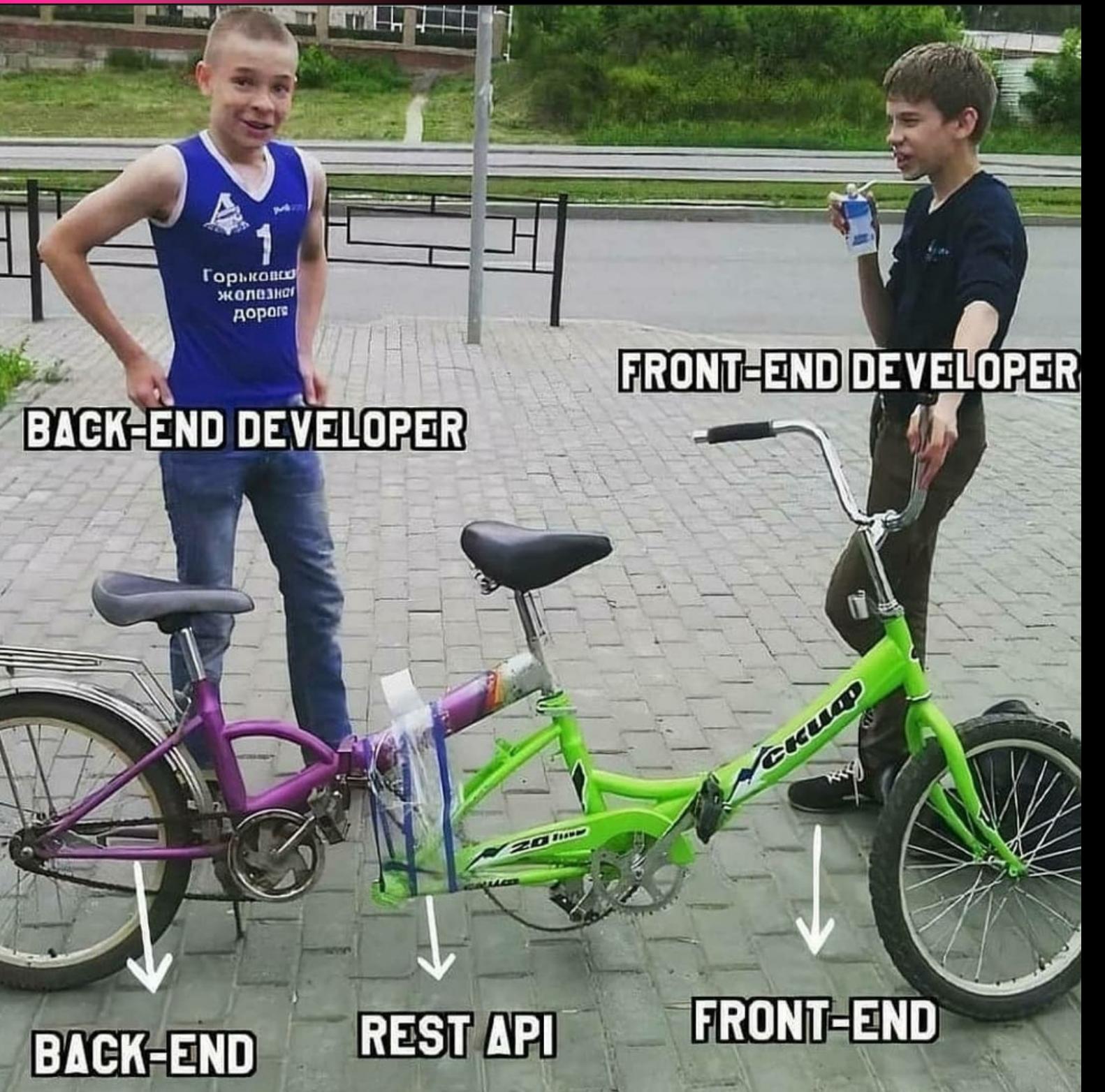
- Why
  - ~ Our requirements
  - ~ Opportunities
- Basics
  - ~ How web works ?
  - ~ APIs
  - ~ endpoints
  - ~ queries
  - ~ request / response
- Two ways to do that
  - ~ API Calls
  - ~ Rendering templates using frameworks

- Calling APIs
  - ~ Backend / API
  - ~ REST Architecture
  - ~ Fetching API
  - ~ Popular frameworks
- Workflow
  - ~ Database Design
  - ~ ORM
  - ~ Server
  - ~ Authentication ( JWT )
  - ~ Frontend
- Rendering Templates
  - ~ Template Engine
  - ~ jinja

# LEVEL UP

- Git ✓
- Programming Languages ✓
- Node ✓
- Django ✓
- Databases ✓
- ORM ✓
- React ( Vue, Next.js, Nuxt.js ) ✓
- Deployment ✓
- Caching
- Testing
- Pipelines
- CI / CD
- Devops
- Docker Containerization
- AWS / GCP





HTTP 200 OK

Allow: GET, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
[  
  {  
    "id": 1,  
    "title": "First blog post",  
    "content": "Hello world!",  
    "created_at": "2018-03-12T20:54:39.360653Z",  
    "updated_at": "2018-03-12T20:54:39.360681Z"  
  },  
  {  
    "id": 2,  
    "title": "Second post",  
    "content": "Building APIs with Django is great.",  
    "created_at": "2018-03-12T20:54:52.301143Z",  
    "updated_at": "2018-03-12T20:54:52.301170Z"  
  },  
  {  
    "id": 3,  
    "title": "3rd post",  
    "content": "Still making posts here.",  
    "created_at": "2018-03-12T20:55:02.620104Z",  
    "updated_at": "2018-03-12T20:55:02.620136Z"  
  }]  
]
```



Endpoints  
Db Queries  
Discuss APIs  
JSON  
Run Postman  
Request/ Response

<https://apiamu.ac.in/api/v1/home-events?lang=en>

[https://apiamu.ac.in/api/v1/amu-news?  
lang=en&year=2012&month=10](https://apiamu.ac.in/api/v1/amu-news?lang=en&year=2012&month=10)

User



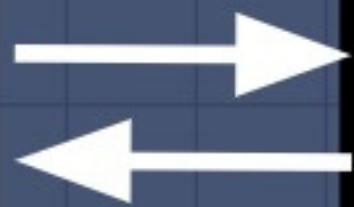
Access  
web app



Frontend



Connect  
via API



Server



Query & Receive  
the data

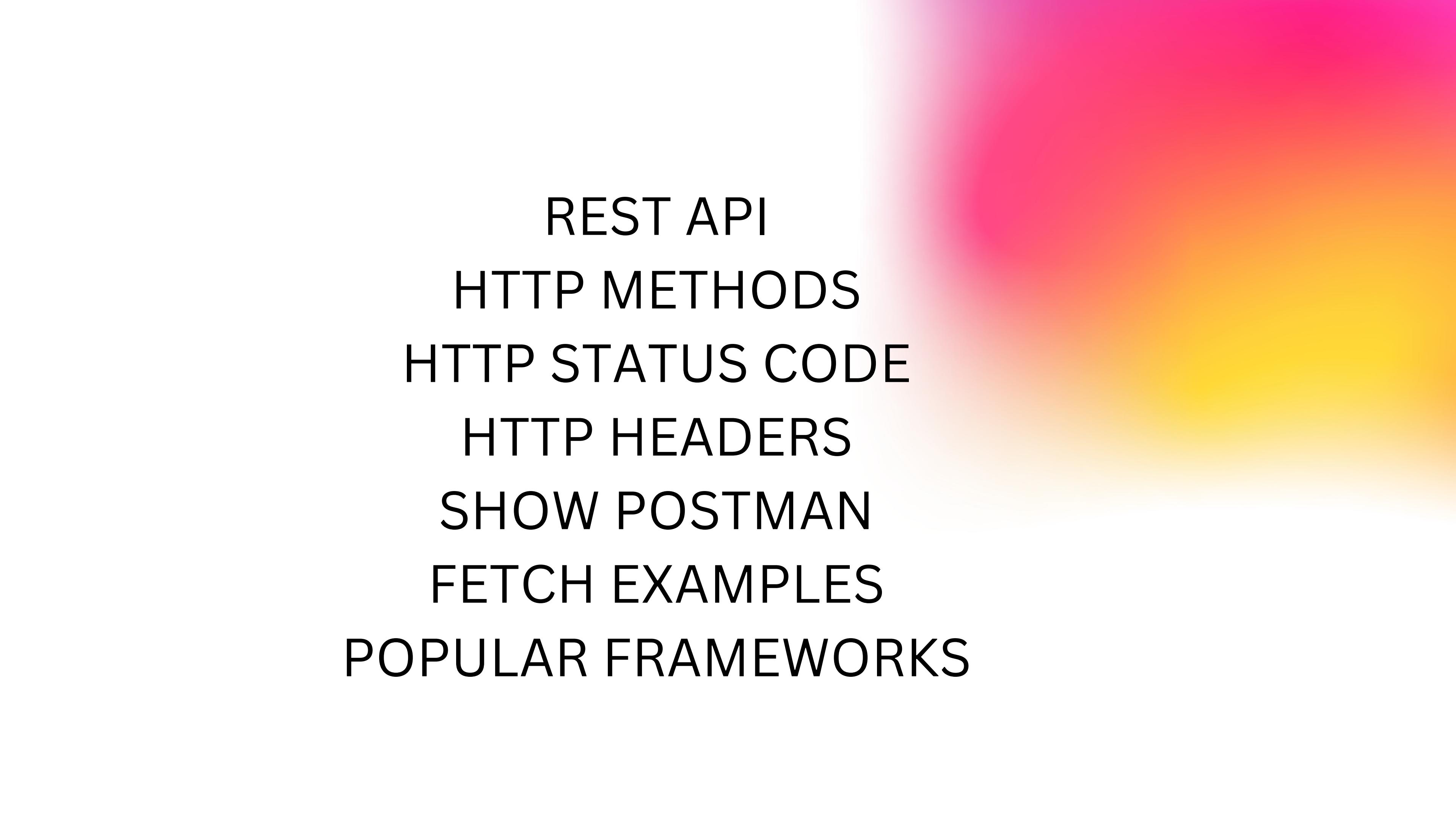


Load data  
from file



```
4
5  from flask import Flask, render_template
6
7  app = Flask(__name__)
8
9  @app.route("/")
10 def home():
11     return render_template("home.html")
12
13 @app.route("/about")
14 def about():
15     return render_template("about.html")
16
17 if __name__ == "__main__":
18     app.run(debug=True)
```

```
1 var express = require('express');
2 var app = express();
3
4 var mustacheExpress = require('mustache-express');
5 app.engine('html', mustacheExpress());
6 app.set('view engine', 'html');
7 app.set('views', __dirname + '/views');
8
9 app.get('/', function(req, res) {
10   res.render('hello.html', {"yourname": "World"});
11 });
12
13 var port = Number(process.env.PORT || 3000);
14 app.listen(port, function() {
15   console.log("Listening on port " + port + "...");
16 });
```



REST API  
HTTP METHODS  
HTTP STATUS CODE  
HTTP HEADERS  
SHOW POSTMAN  
FETCH EXAMPLES  
POPULAR FRAMEWORKS

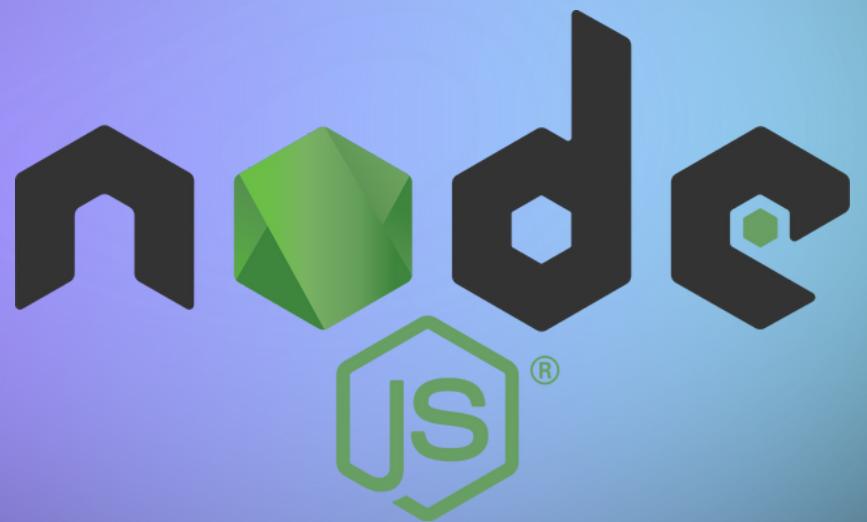
# Workflow of a Blog Application

Database Design  
URLs/ Endpoints

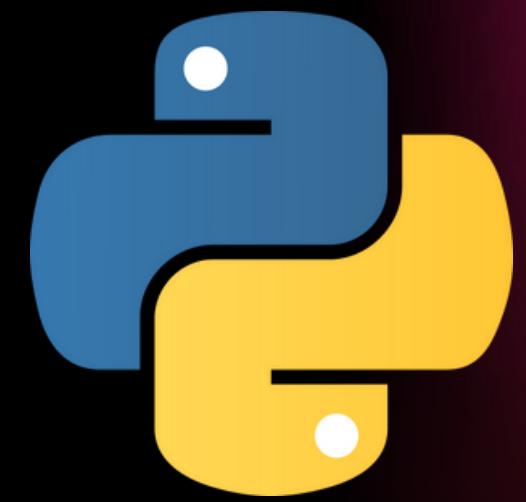
Queries

Response

Authentication  
Middlewares



# Blog API



# django Application