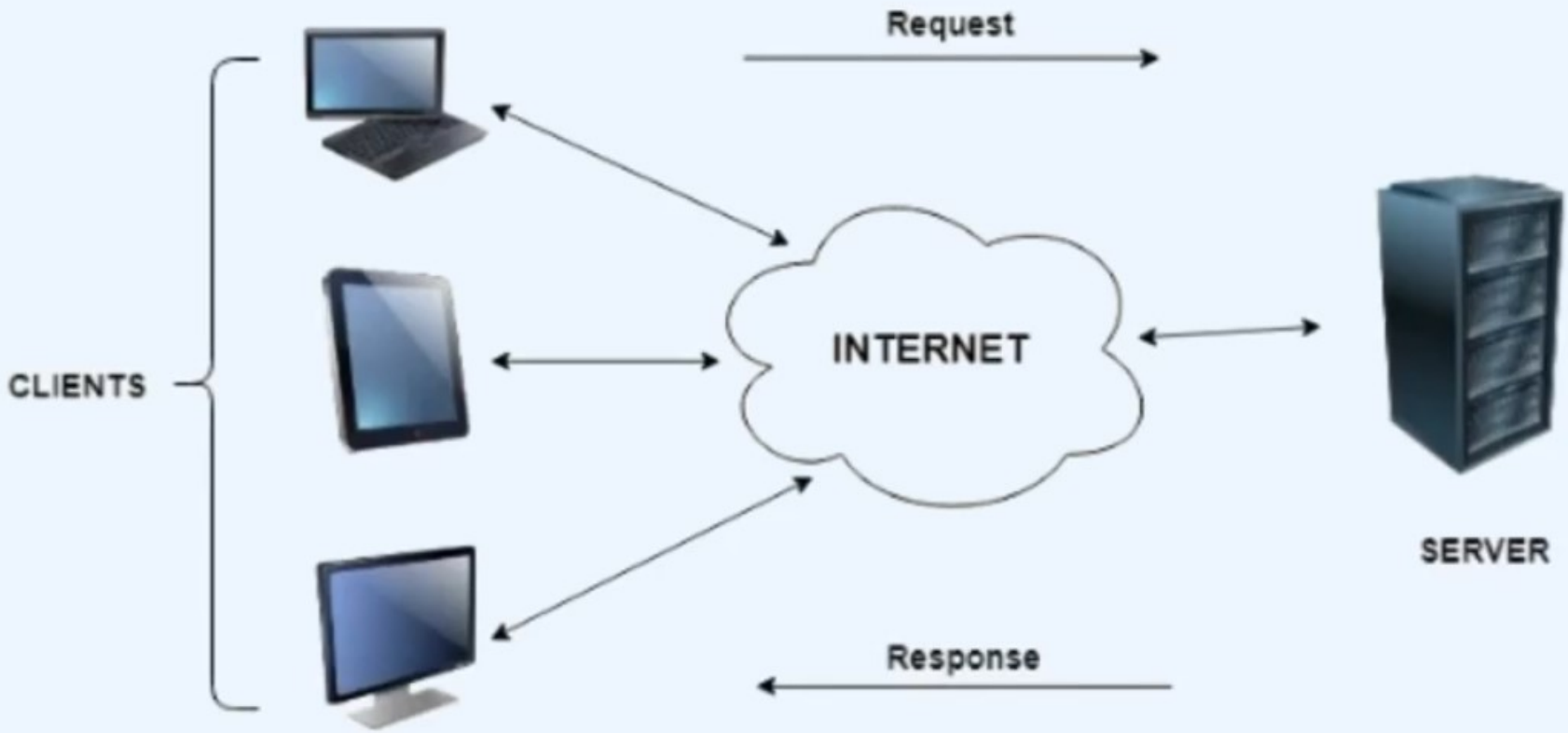
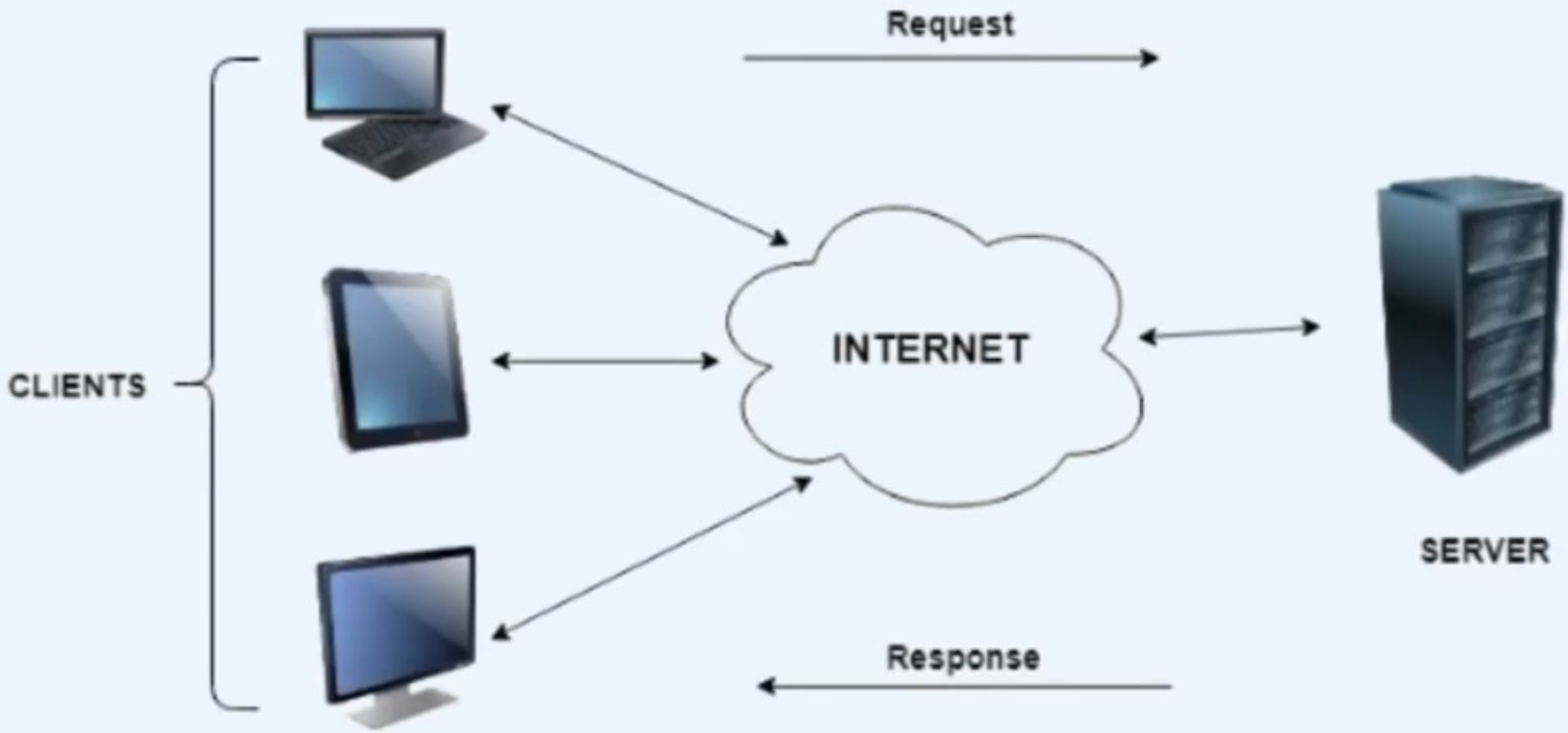
What is a server

arrifat011@gmail.com



What is a server

arrifat011@gmail.com



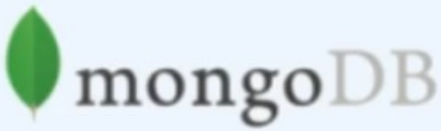
What is node js and Express js and why use them

arrifat011@gmail.com

What is node js and Express js and why use them

arrifat011@gmail.com

Express



- Client and Server Connection with Request and response
- (Recap) install node-express and use fetch to load data
- Create React form and Post API and send data to the server
- Send Client data to the server and display data on the client

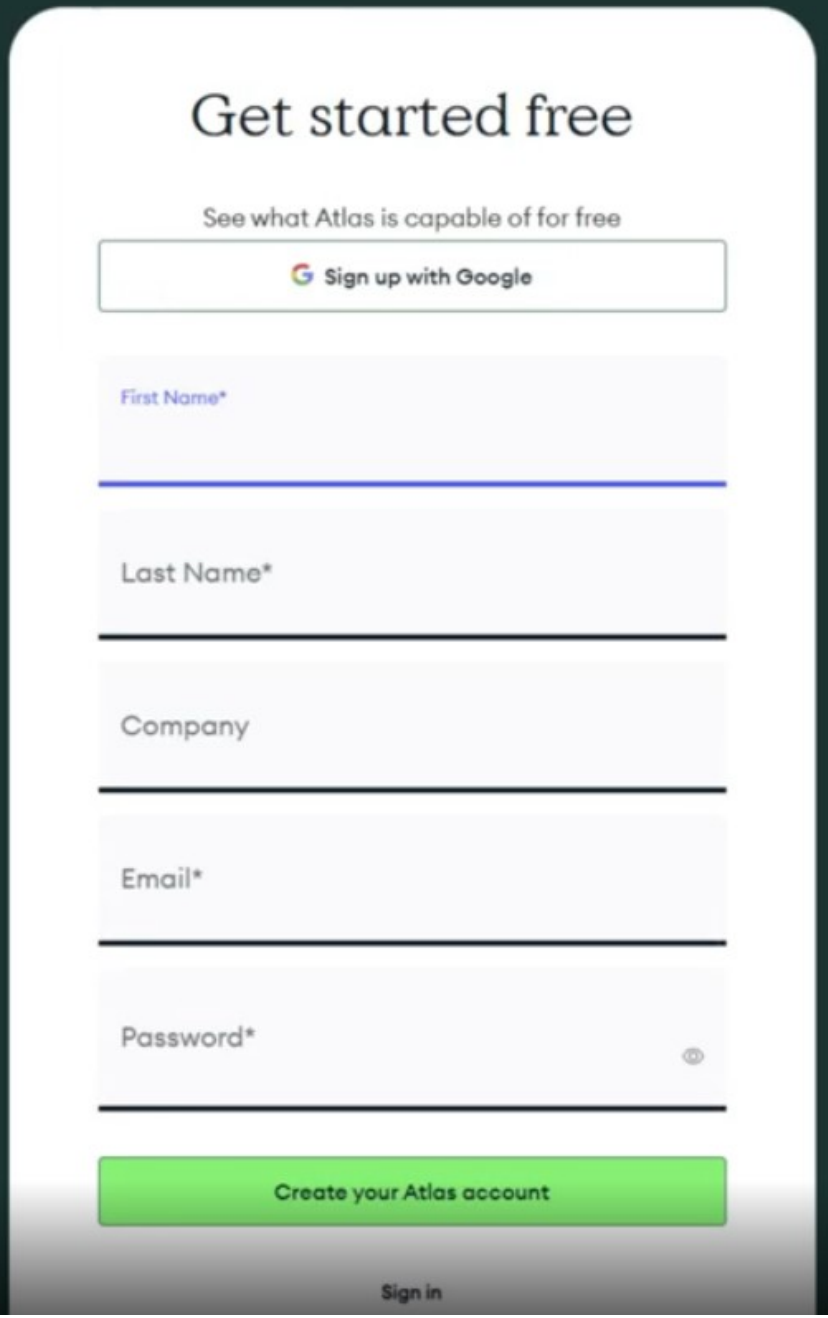
Create mongodb atlas account for simple Crud server

MongoDB Atlas

- **Work with your data as code**
  Documents in MongoDB map directly to objects in your programming language. Modify your schema as your apps grow over time.

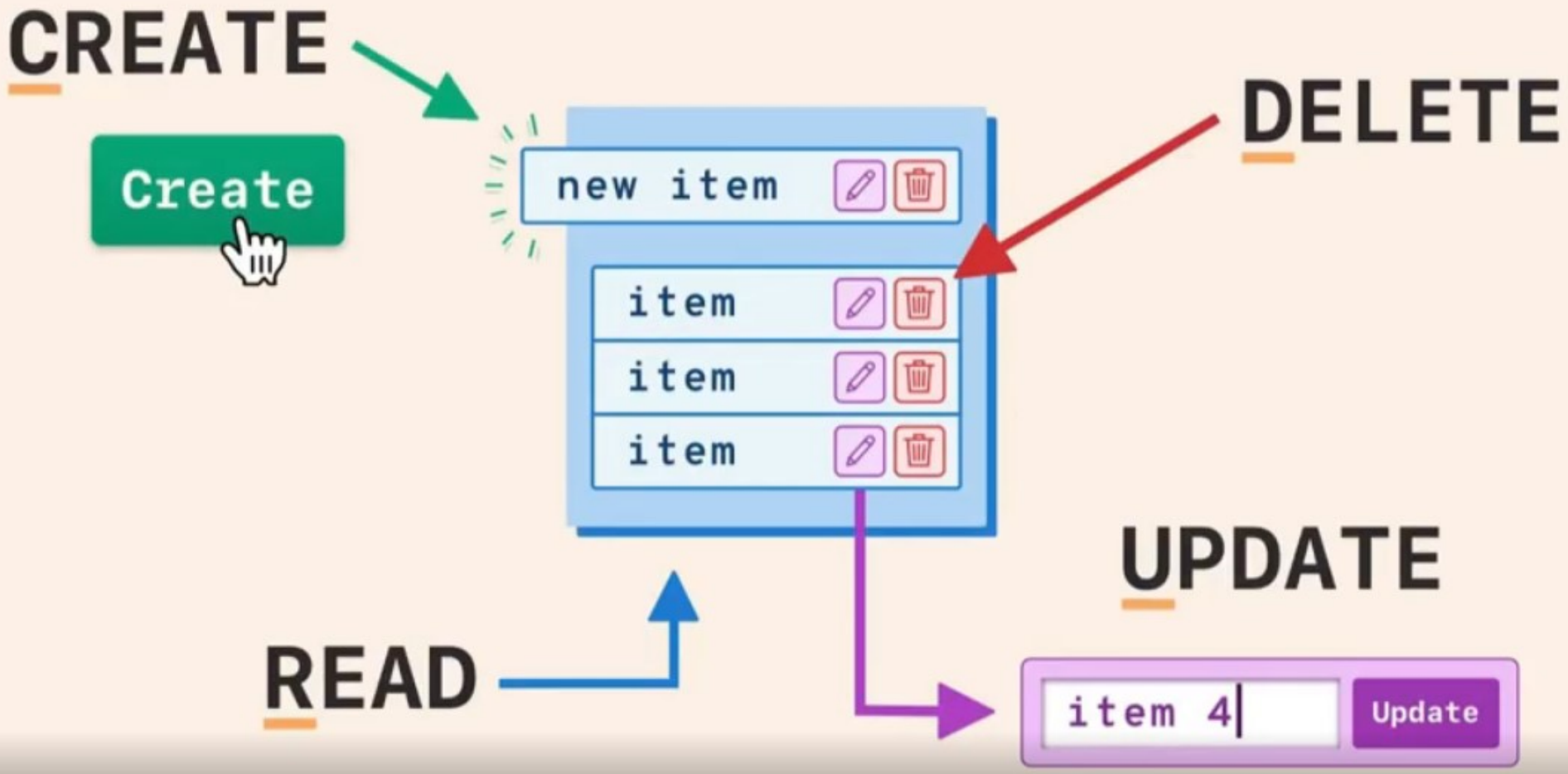
- **Focus on building, not managing**
  Let MongoDB Atlas take care of the infrastructure operations you need for performance at scale, from always-on security to point-in-time recovery.

- **Simplify your data dependencies**
  Leverage application data for full-text search, real-time analytics, rich visualizations and more with a single API and minimal data movement.



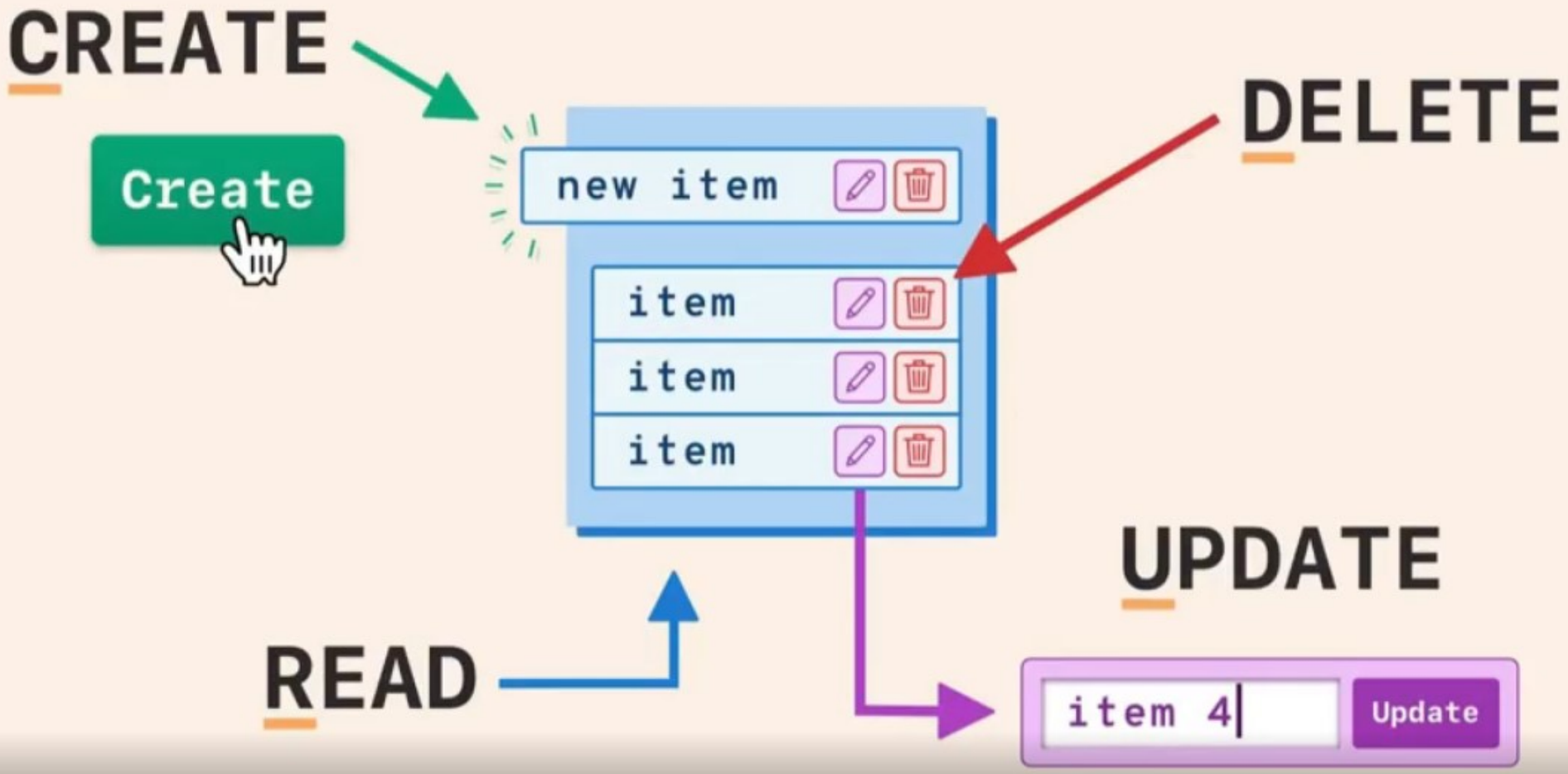
Create Read and delete And Show Remaining Users

arrifat011@gmail.com



Create Read and delete And Show Remaining Users

arrifat011@gmail.com



HOME FEATURES WORK SERVICES BLOG CONTACT

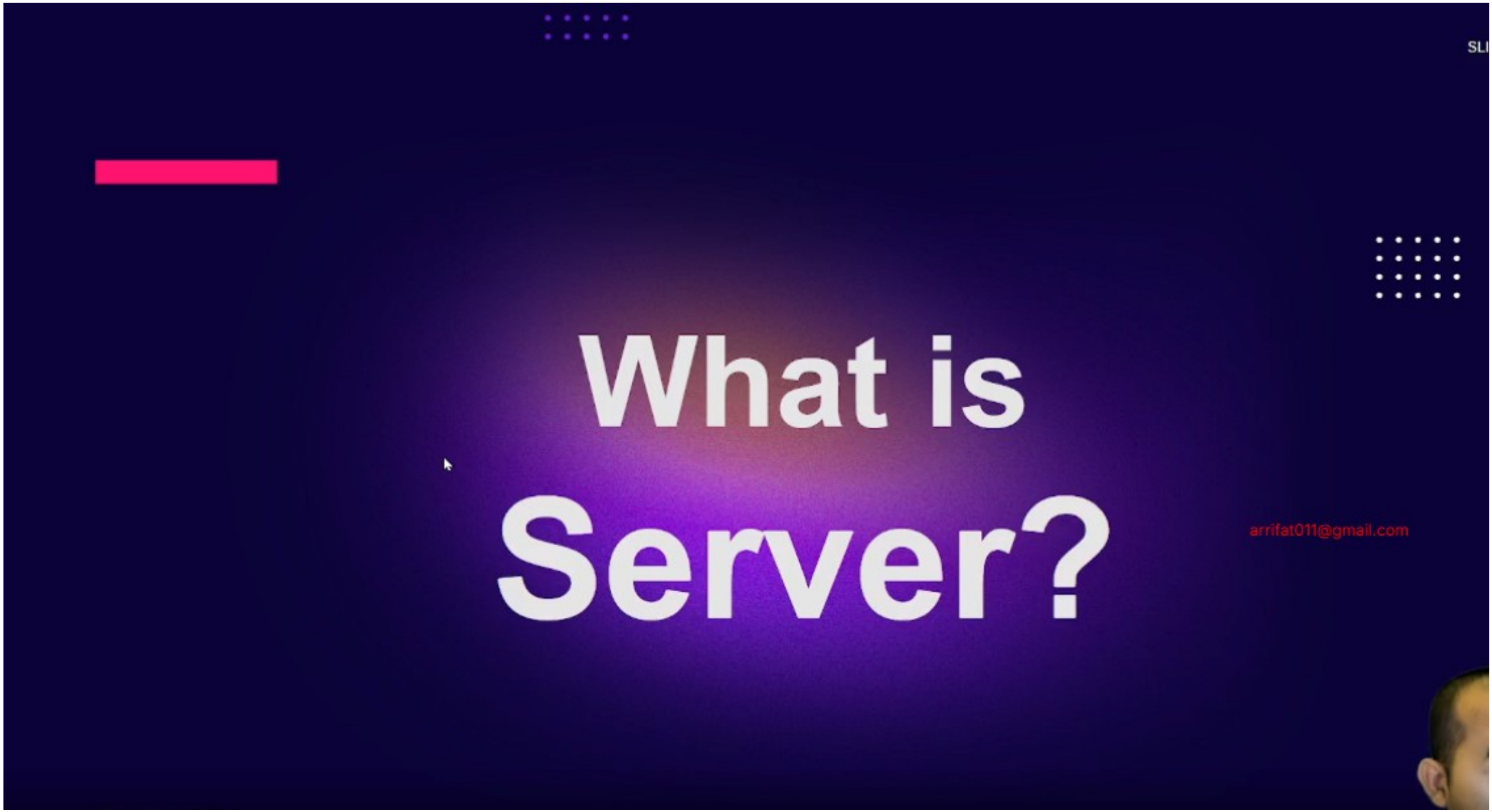
- Module Overview and create basic server and client
- Client Side router and Connect MongoDB database
- Server Side hide database User and Password using dotenv config
- Create Add coffee from Client Side and get data
- Send Coffee data to the server and store in the mongodb database
- View all coffee and create coffee card
- Delete a coffee and fetch coffee for update
- Update single coffee information

arrifat011@gmail.com

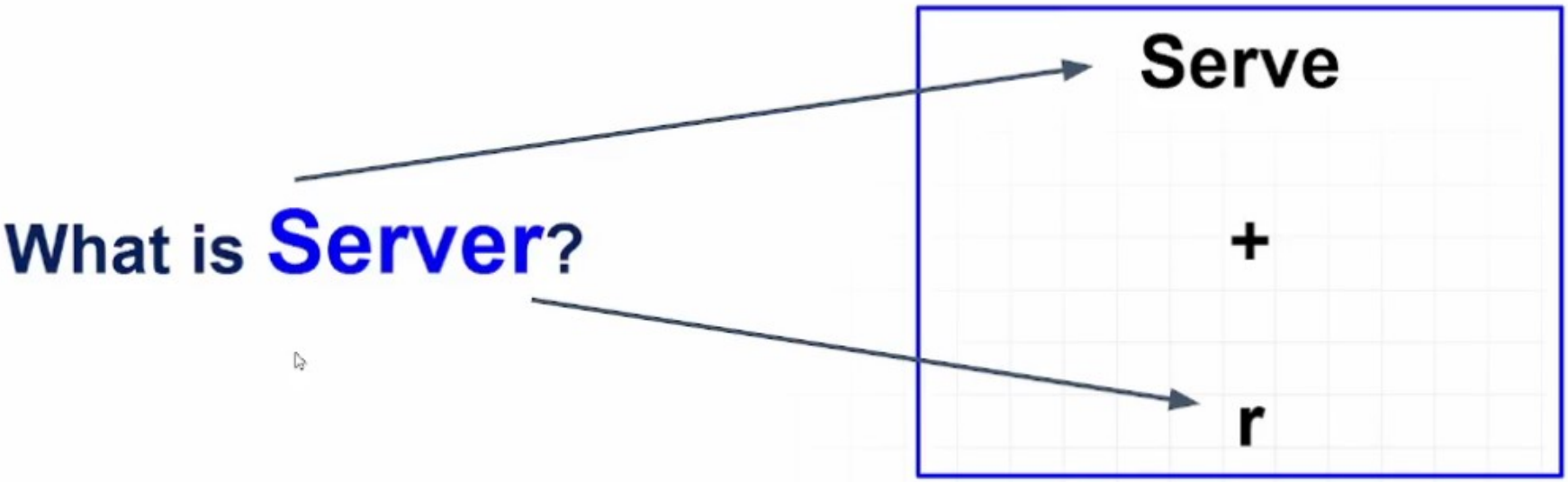
HOME FEATURES WORK SERVICES BLOG CONTACT

- Module Overview and create basic server and client
- Client Side router and Connect MongoDB database
- Server Side hide database User and Password using dotenv config
- Create Add coffee from Client Side and get data
- Send Coffee data to the server and store in the mongodb database
- View all coffee and create coffee card
- Delete a coffee and fetch coffee for update
- Update single coffee information

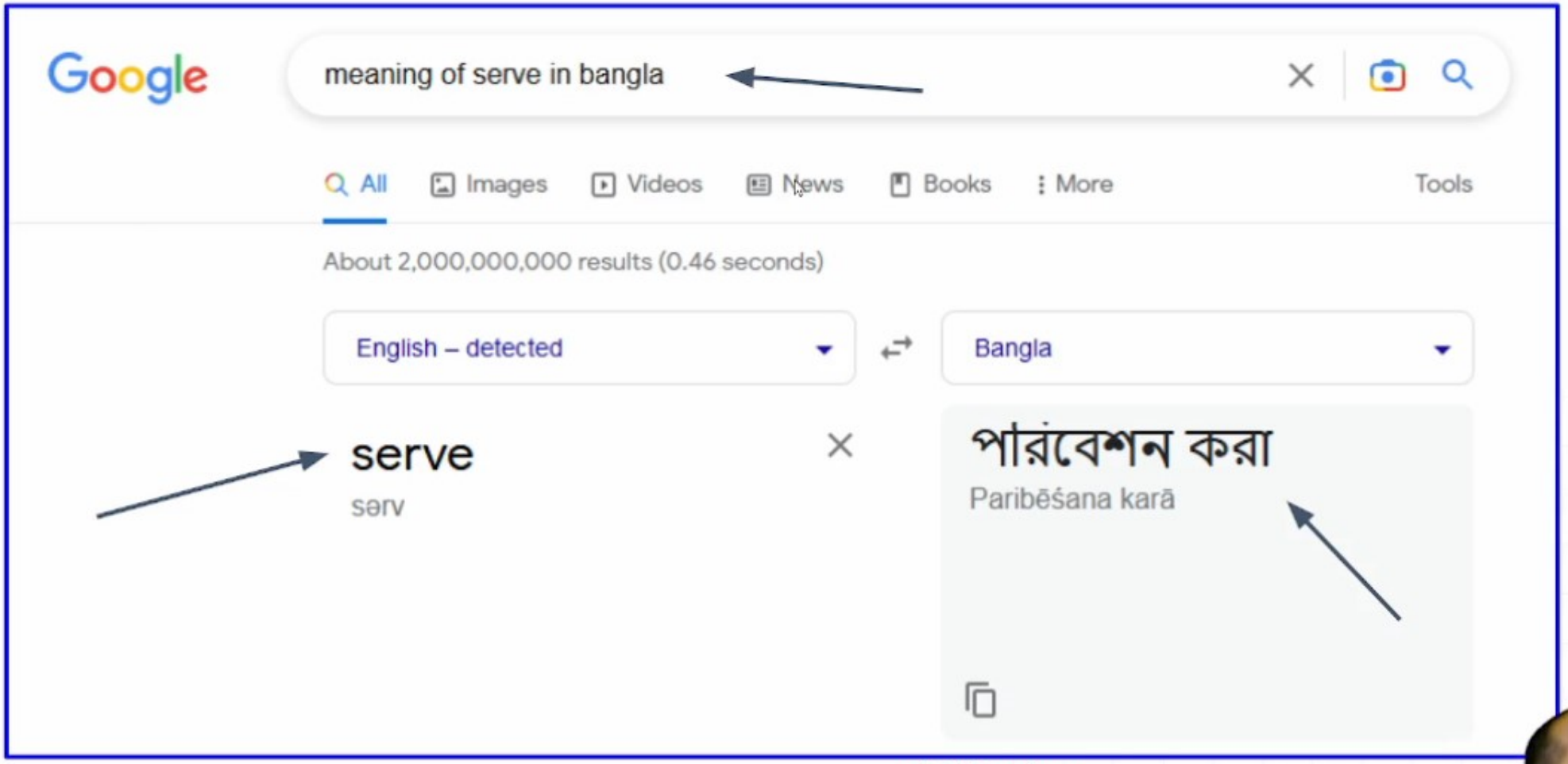
arrifat011@gmail.com


What is
Server?
arrifat011@gmail.com

What is **Server**?



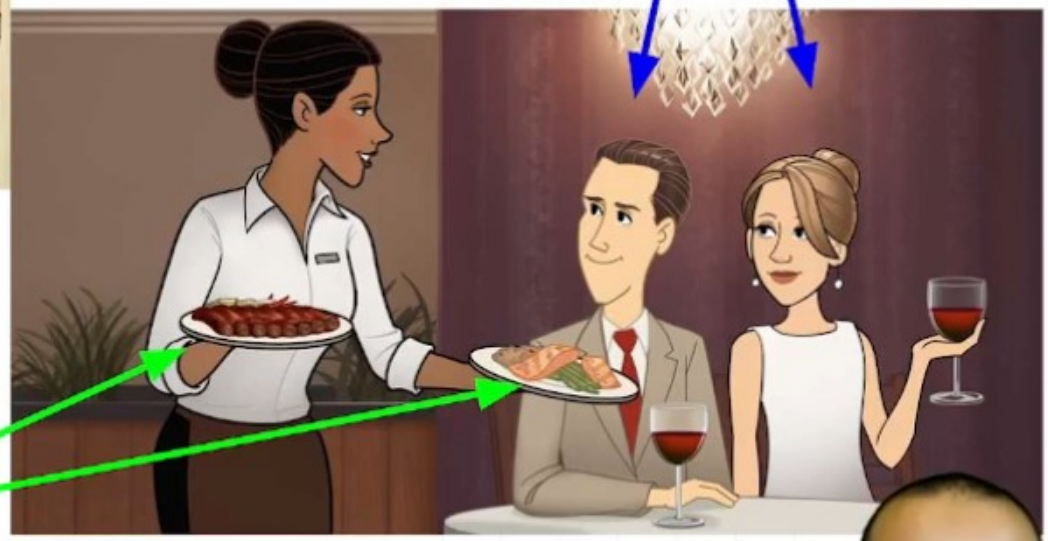
arrifat011@gmail.com



arrifat011@gmail.com

Client

API



Server

Data

Server কি দিয়ে তৈরি?

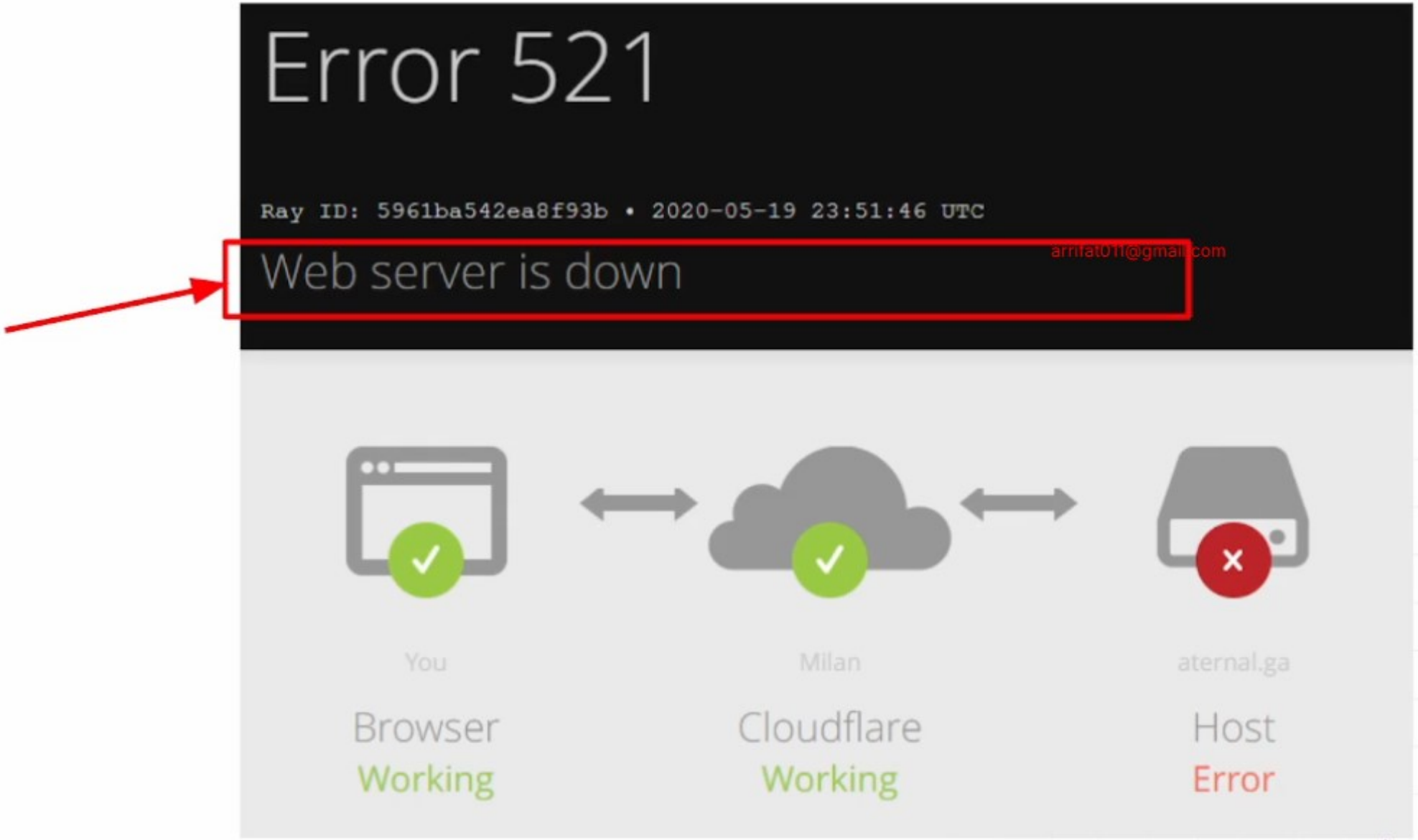
arrifat011@gmail.com

তোমার computer কে চালায়?

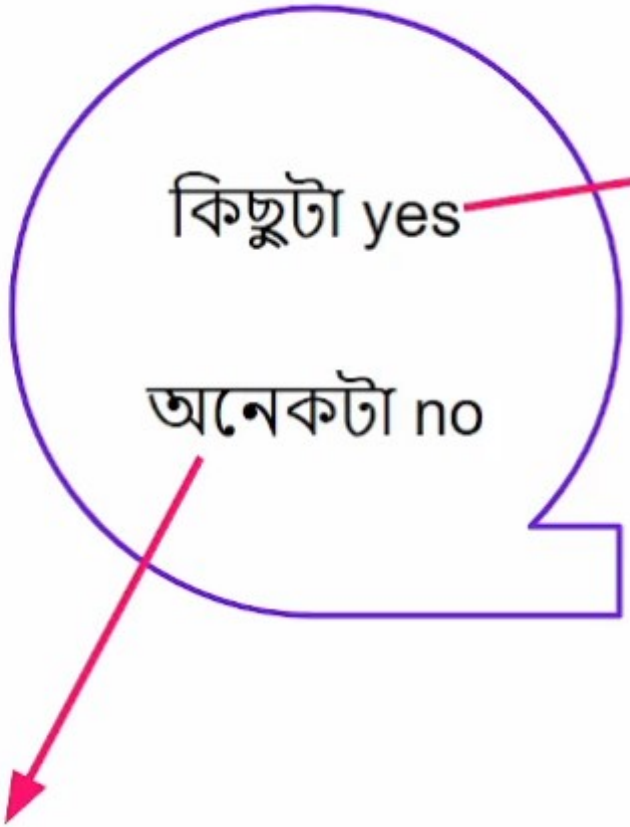
-> Operating system? **Windows? Linux?**

তাহলে Server কে চালায়?

-> একটা অপারেটিং সিস্টেম চালায় যা ২৪/৭ চলতেই থাকে। কারণ off হয়ে গেলেই সে তোমাকে **Serve করতে পারবে না।**


Error 521
Ray ID: 5961ba542ea8f93b • 2020-05-19 23:51:46 UTC
Web server is down
arrifat011@gmail.com
You
Browser
Working
Milan
Cloudflare
Working
aternal.ga
Host
Error

arrifat011@gmail.c

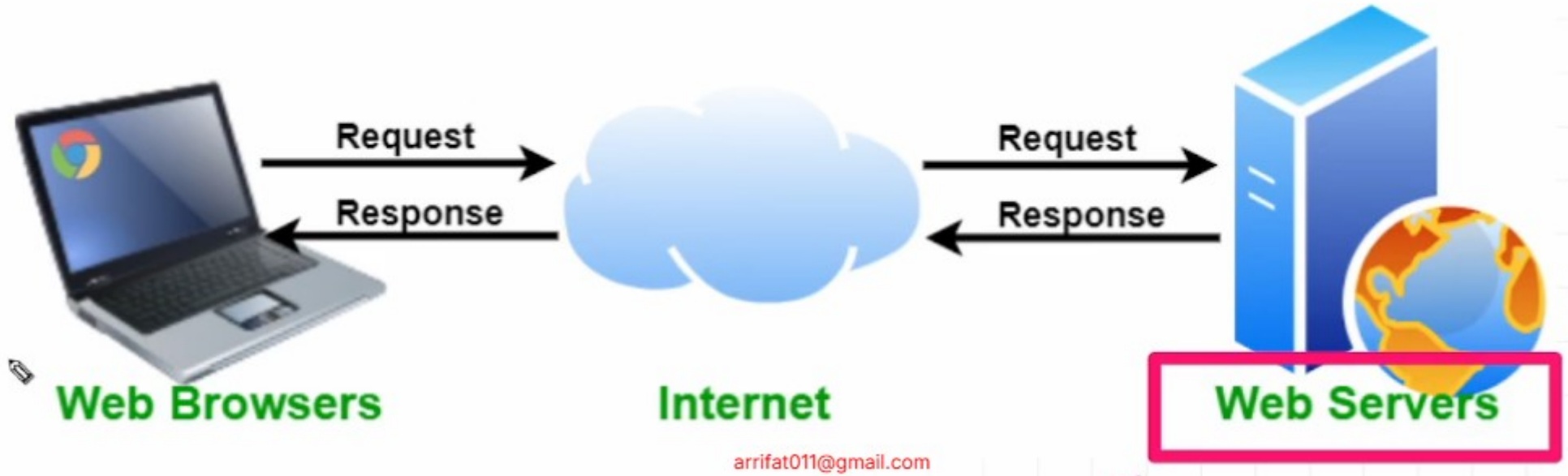


How???

Why???

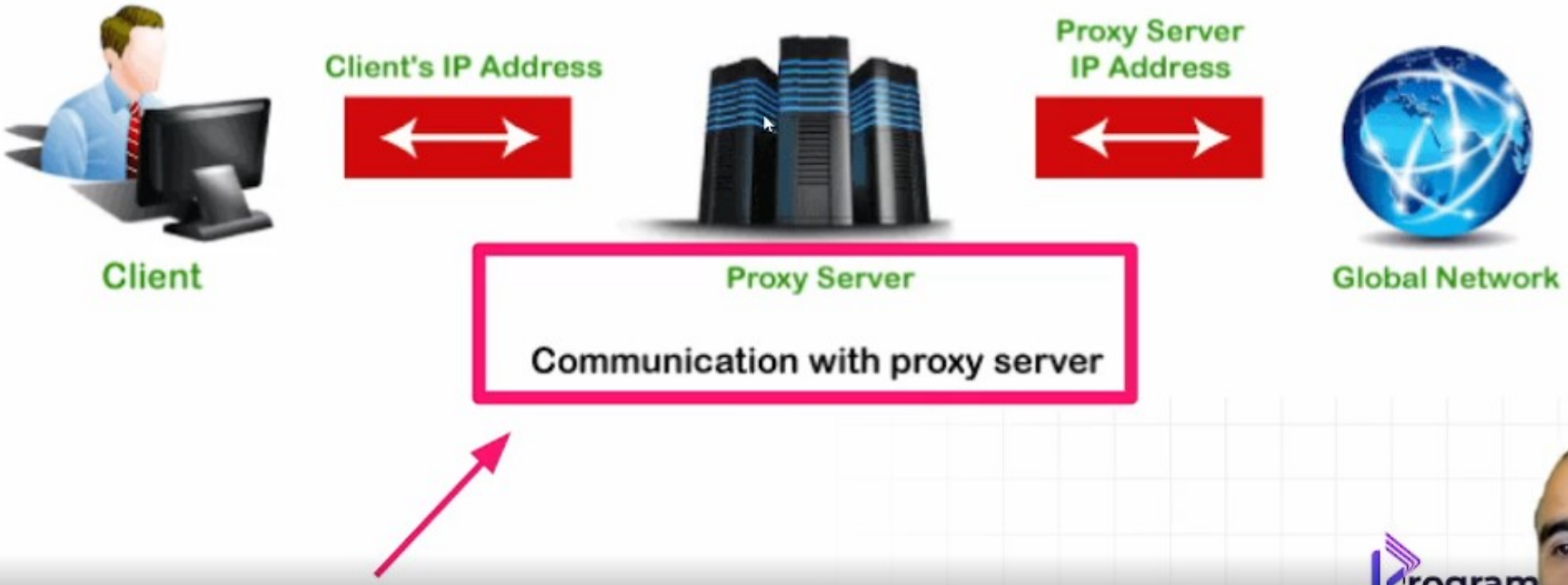
Server কি আমাদের personal কম্পিউটার এর মত?

Example of server

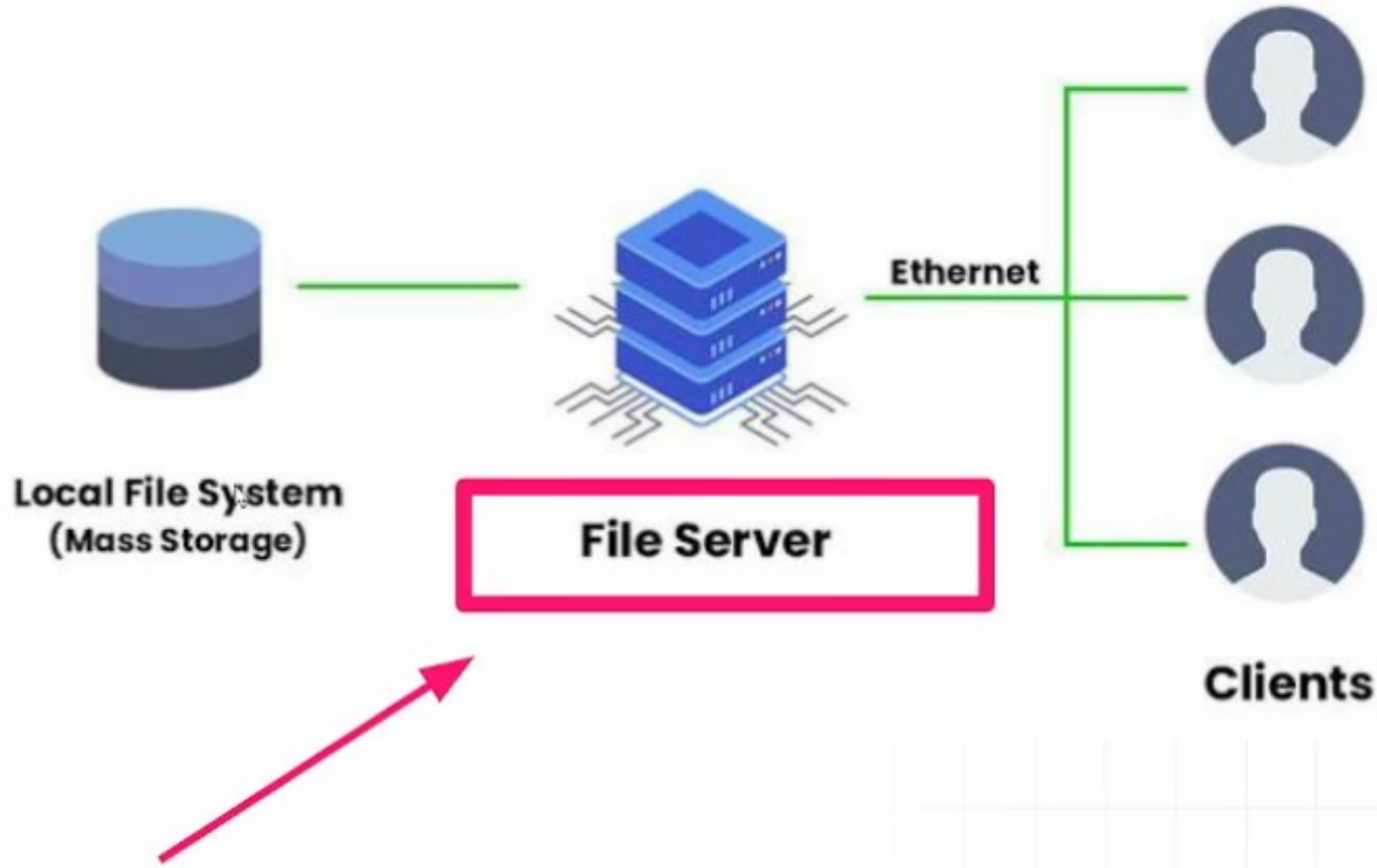


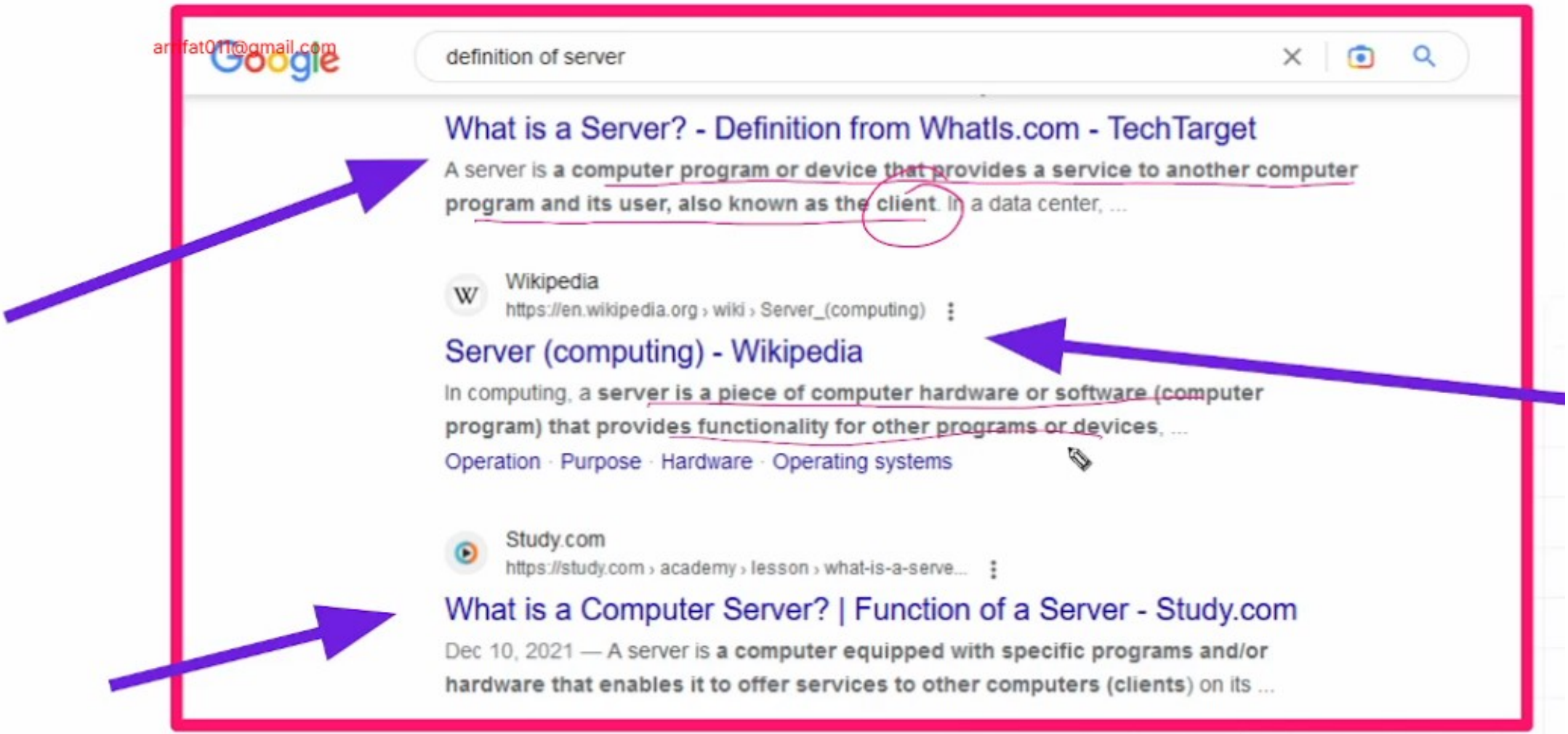
arrifat011@gmail.com

Example of server



Example of server

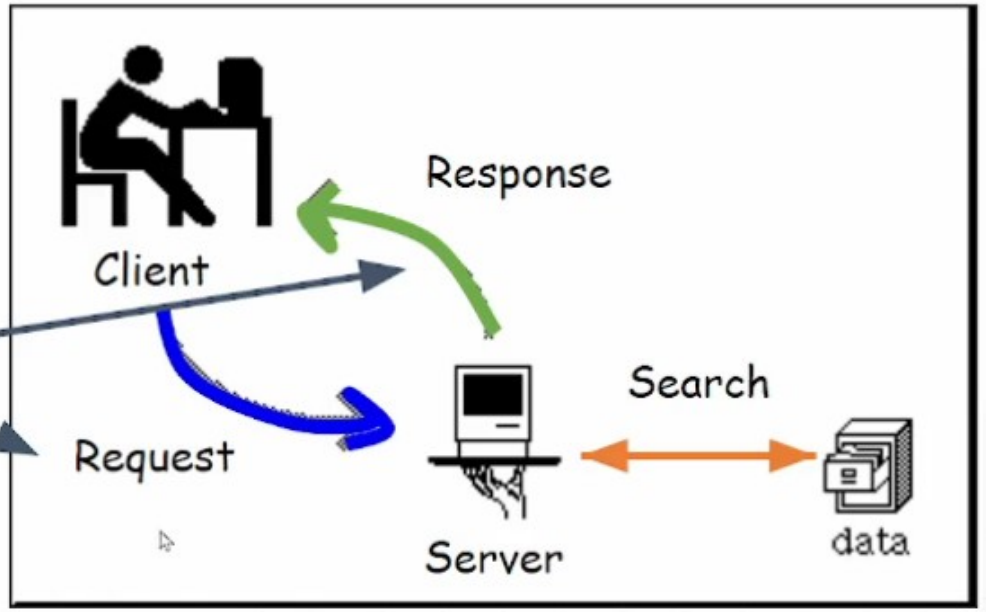


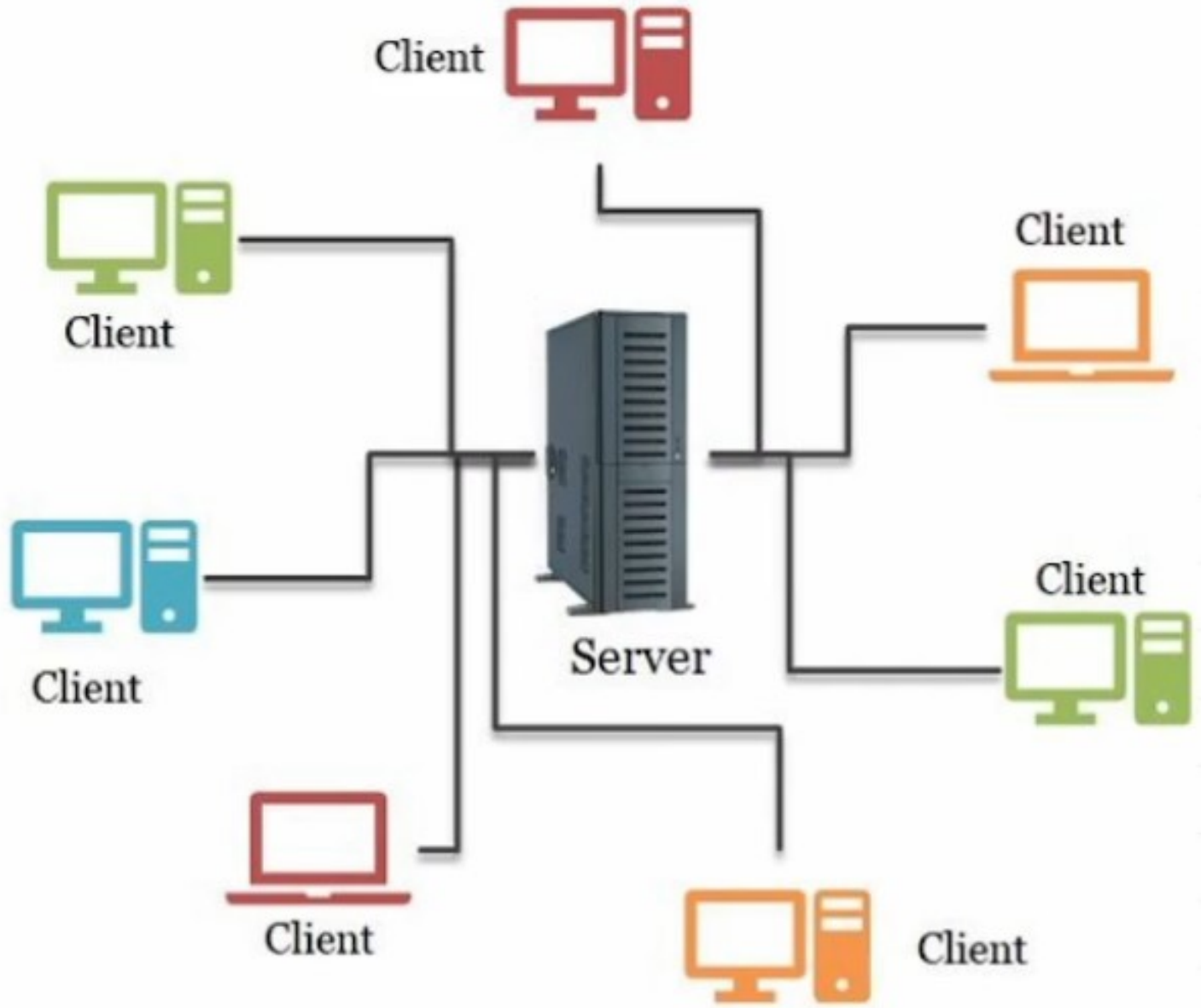

arifat011@gmail.com
Google
definition of server
What is a Server? - Definition from WhatIs.com - TechTarget
A server is a computer program or device that provides a service to another computer program and its user, also known as the client. In a data center, ...
Wikipedia
https://en.wikipedia.org › wiki › Server_(computing)
Server (computing) - Wikipedia
In computing, a server is a piece of computer hardware or software (computer program) that provides functionality for other programs or devices, ...
Operation · Purpose · Hardware · Operating systems
Study.com
https://study.com › academy › lesson › what-is-a-serve...
What is a Computer Server? | Function of a Server - Study.com
Dec 10, 2021 — A server is a computer equipped with specific programs and/or hardware that enables it to offer services to other computers (clients) on its ...

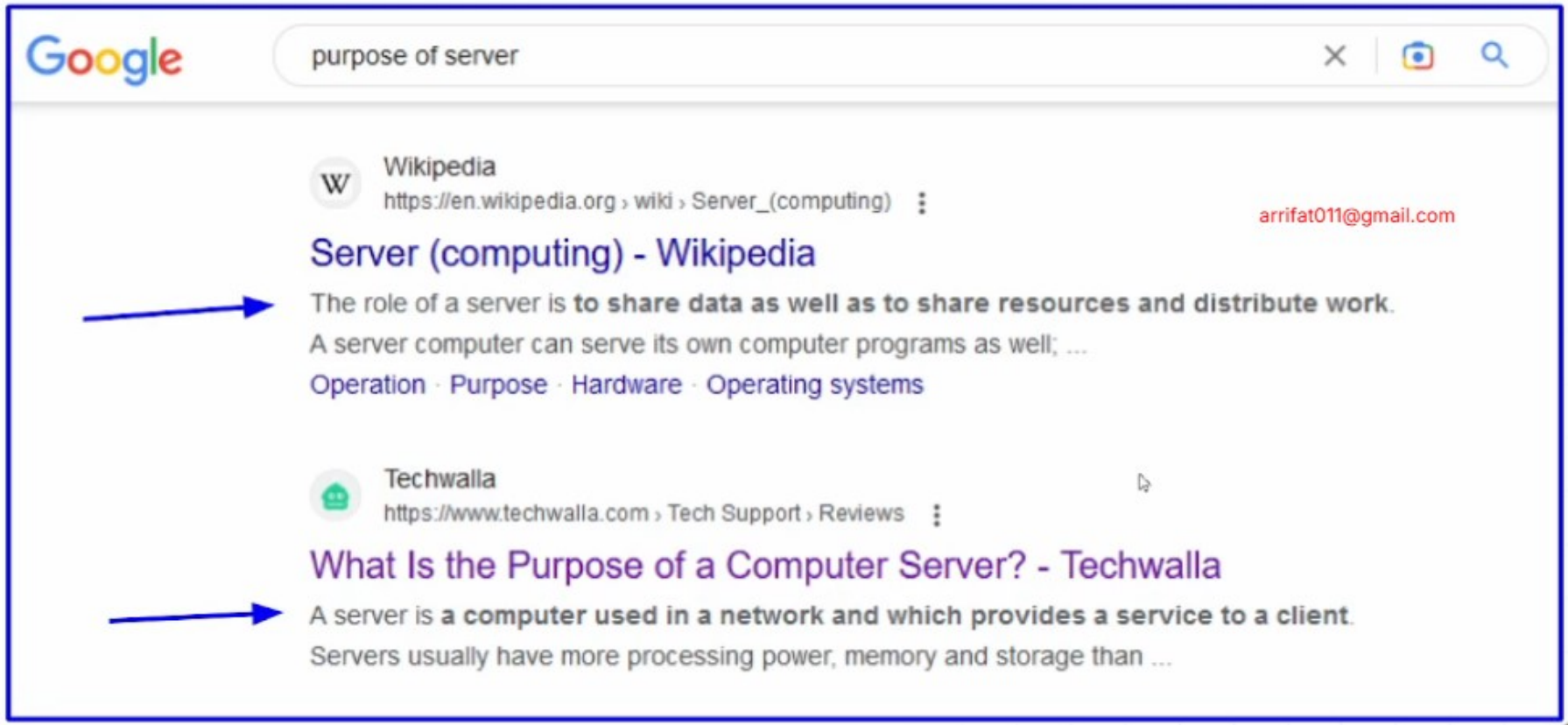
arrifat011@gmail.com

Request নেওয়া

Response দেওয়া



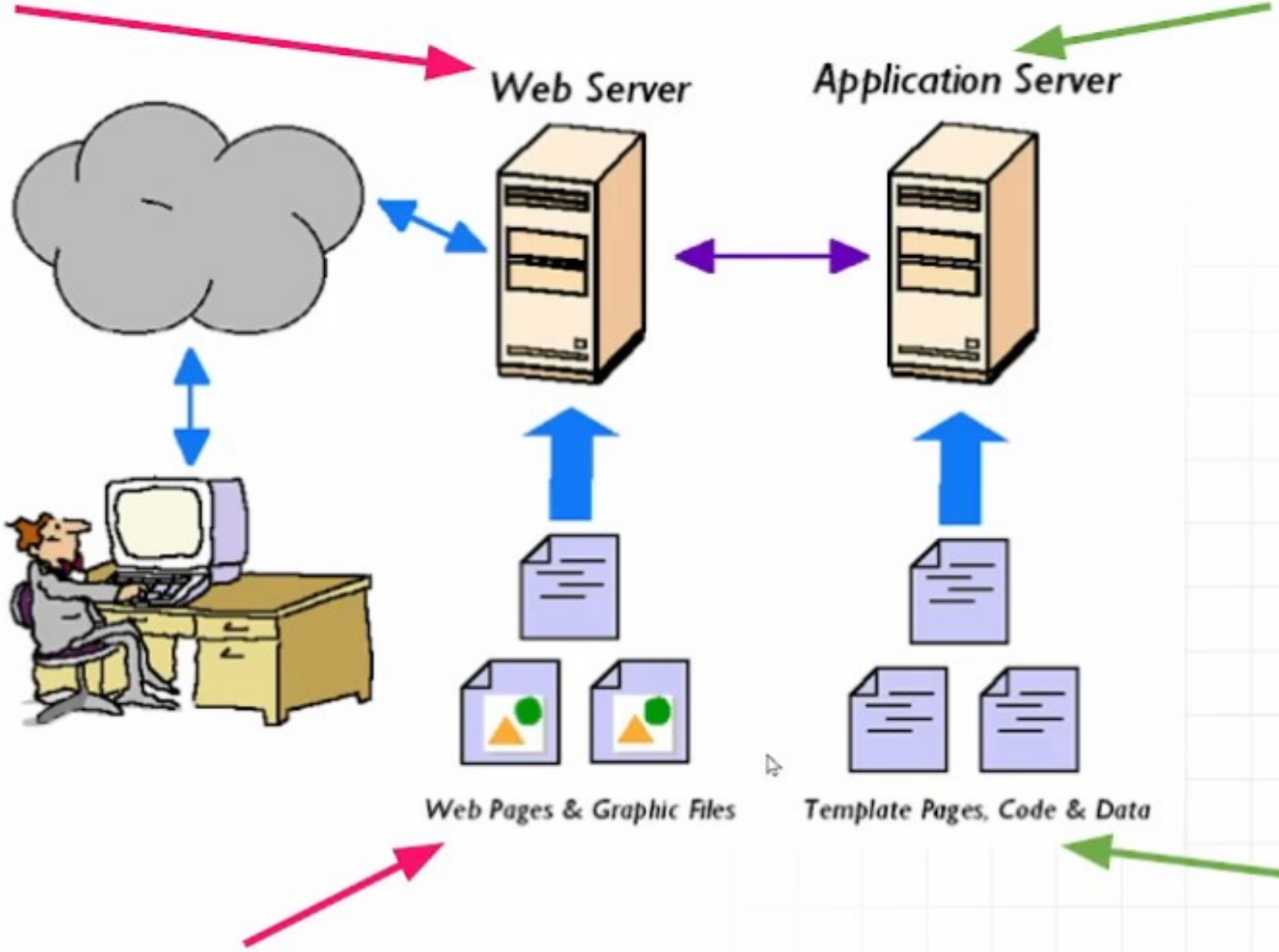

Client
Client
Client
Client
Server
Client
Client
Client


Google
purpose of server
Wikipedia
https://en.wikipedia.org › wiki › Server_(computing)
arrifat011@gmail.com
Server (computing) - Wikipedia
The role of a server is to share data as well as to share resources and distribute work.
A server computer can serve its own computer programs as well; ...
Operation · Purpose · Hardware · Operating systems
Techwalla
https://www.techwalla.com › Tech Support › Reviews
What Is the Purpose of a Computer Server? - Techwalla
A server is a computer used in a network and which provides a service to a client.
Servers usually have more processing power, memory and storage than ...

সব ধরনের server এর কাজ একই রকম?

No

arrifat011@gmail.com


Web Server
Application Server
Web Pages & Graphic Files
Template Pages, Code & Data

Web server: Hosting and serving websites to clients over the internet.

File server: Sharing and storing files over a network.

Application server: Hosting & running applications for many users to access remotely.

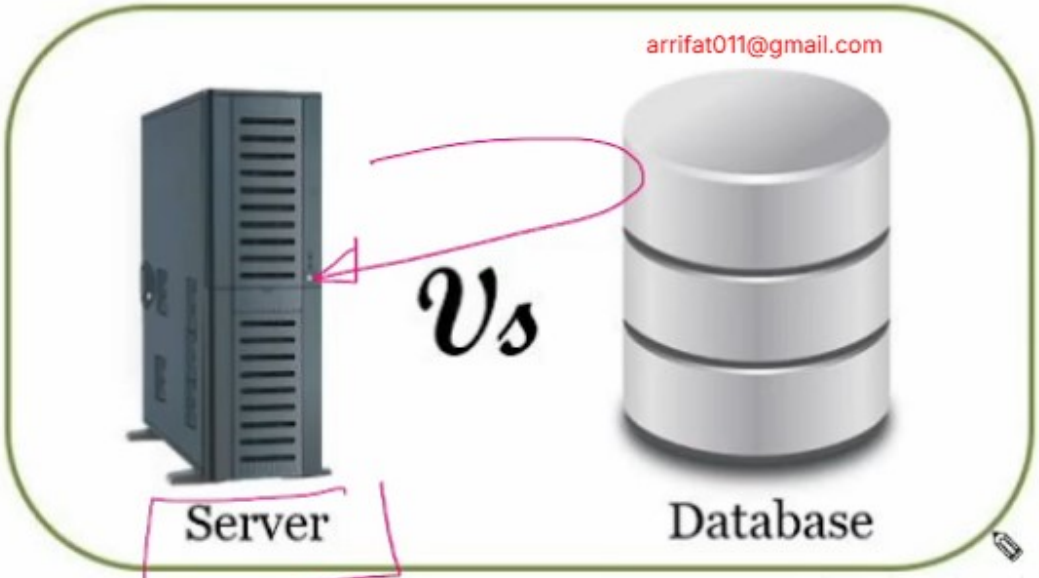
Database server: Storing, managing and serving databases to multiple clients.

Server & Database

দুইটা কি একই জিনিস?

A computer program that runs to provide services.

arrifat011@gmail.com



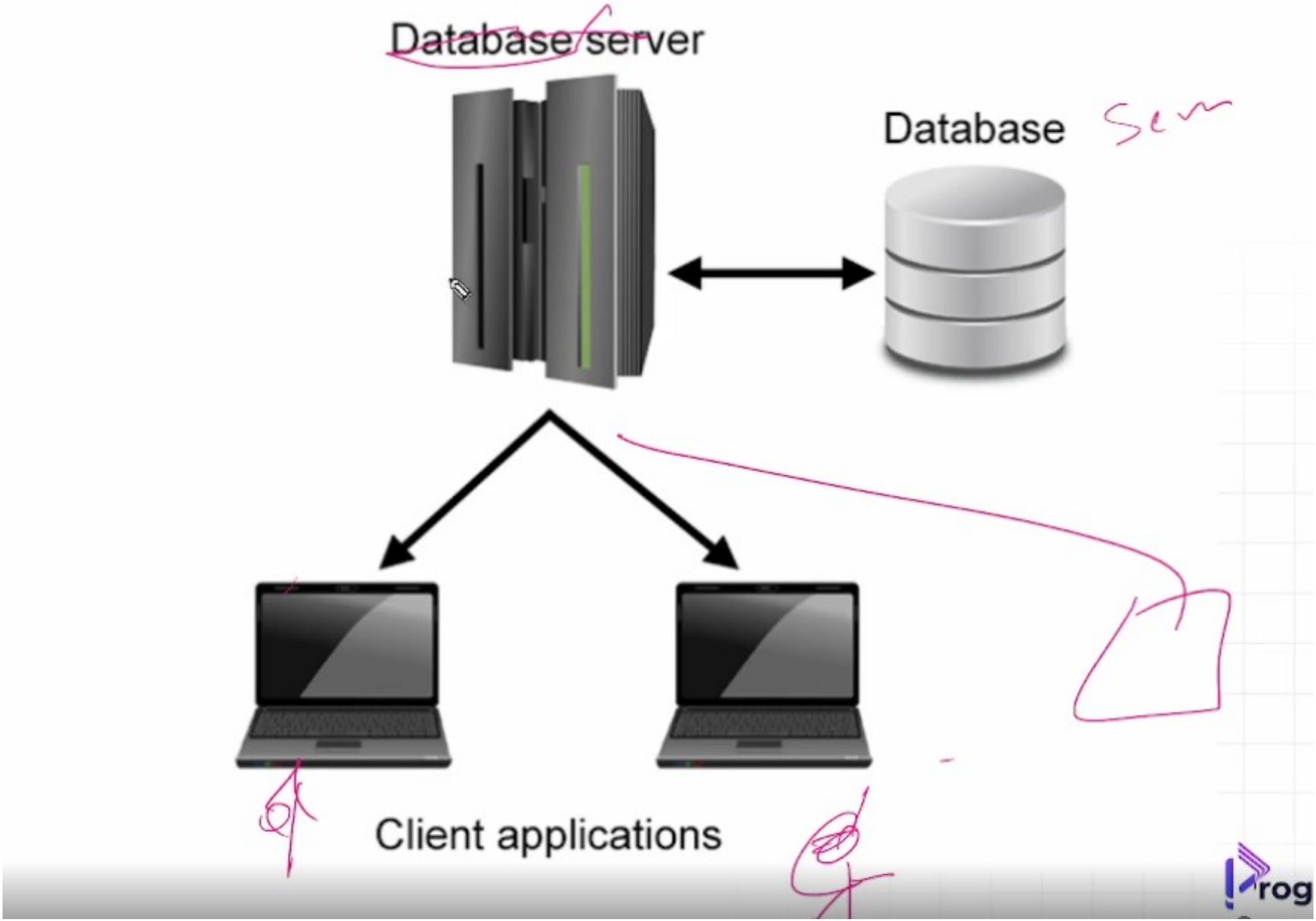
Large collection of different types of data

Relation between

client, server & database

???

arrifat011@gmail.com


Database server
Database
Client applications

Advantages of Server

Reduce physical resource cost

Speed setup

Create backups

Gain Flexibility

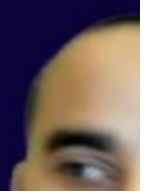
Advantages of Server

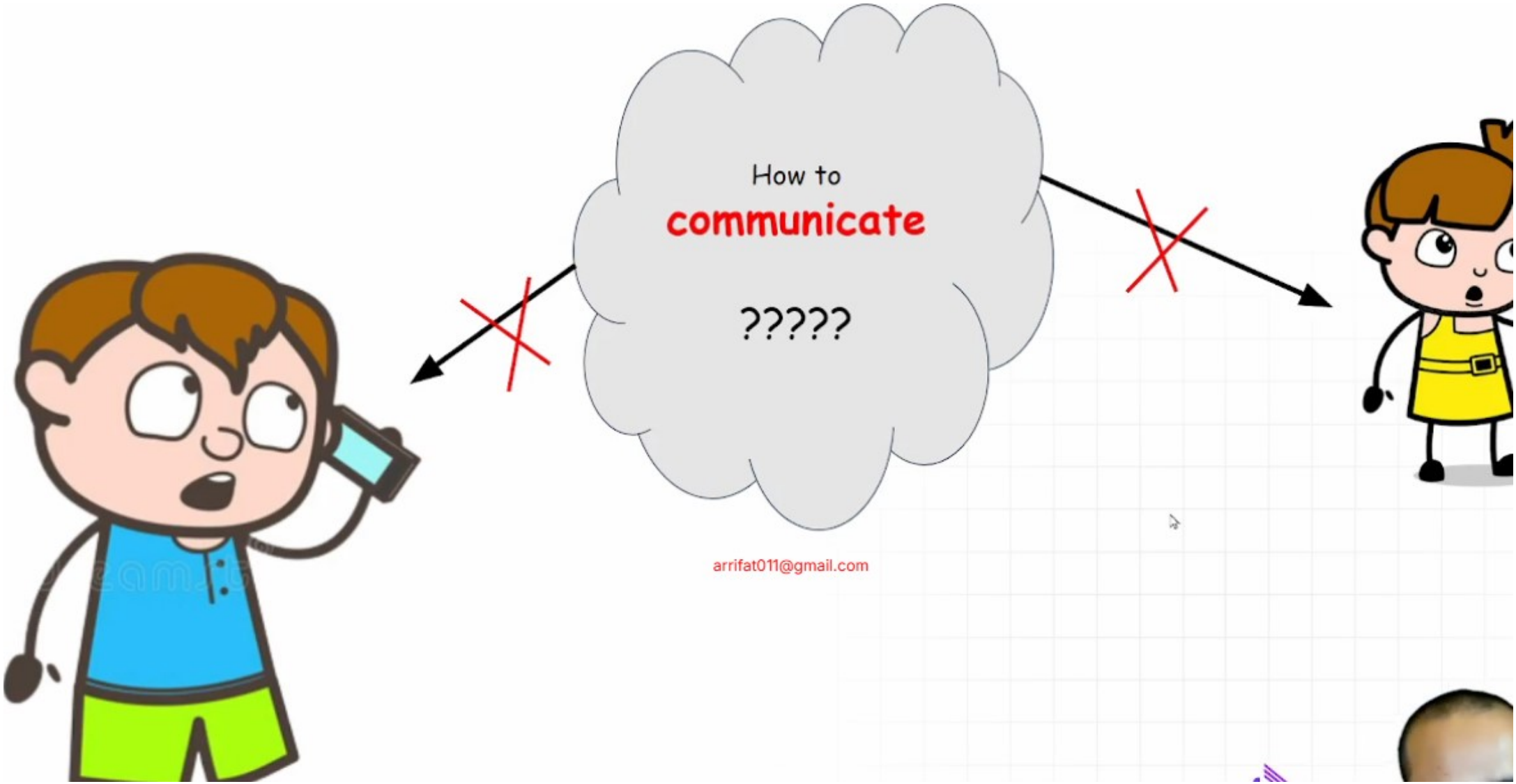
arrifat011@gmail.com

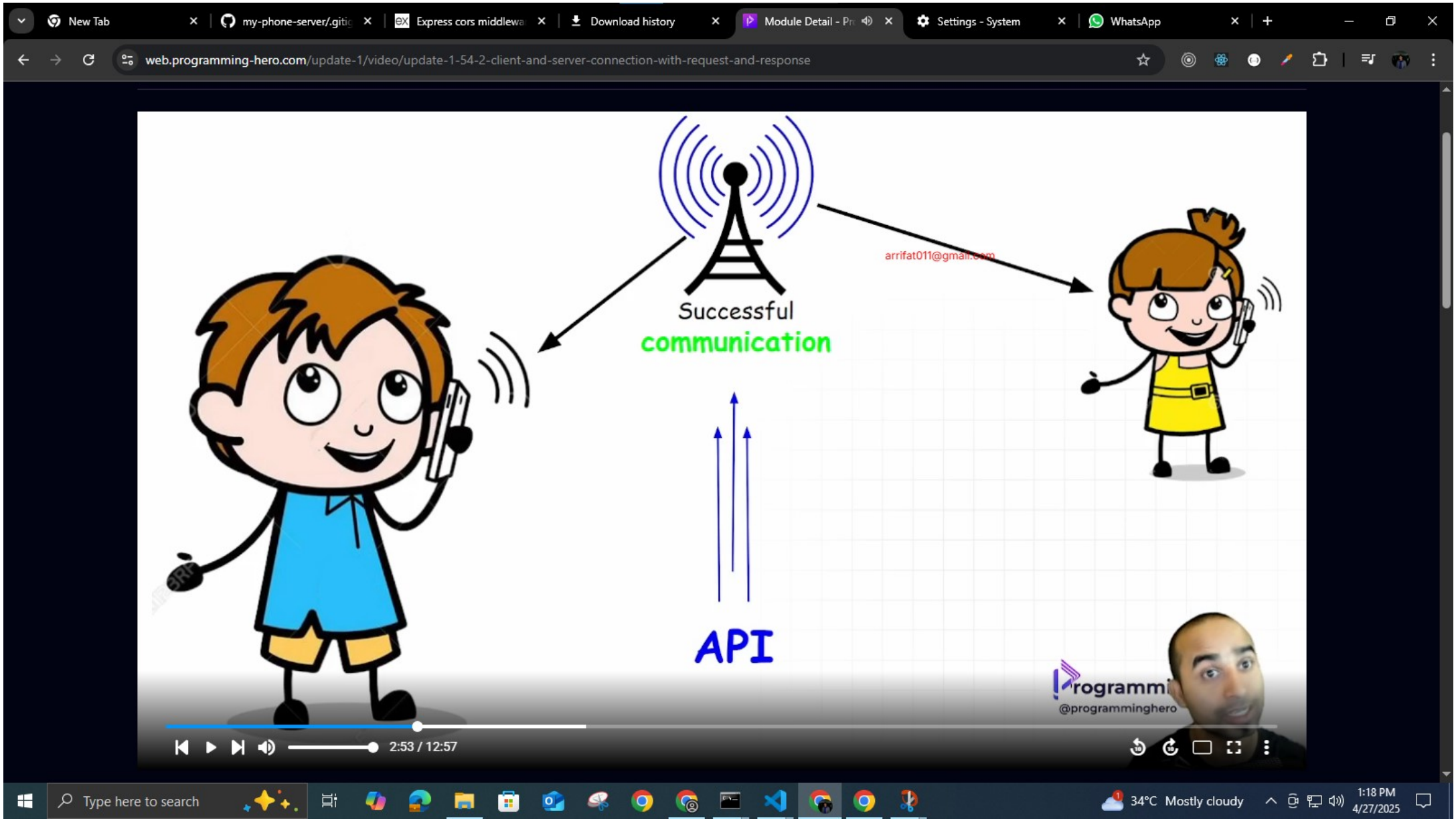
- Centrally controls the clients connected to a network.
- Dedicated server for every work.
- Scalable.
- Facilitates easy backup and recovery of data.
- Remotely access the servers using various platforms.

arrifat011@gmail.com

How does API connect client & server?

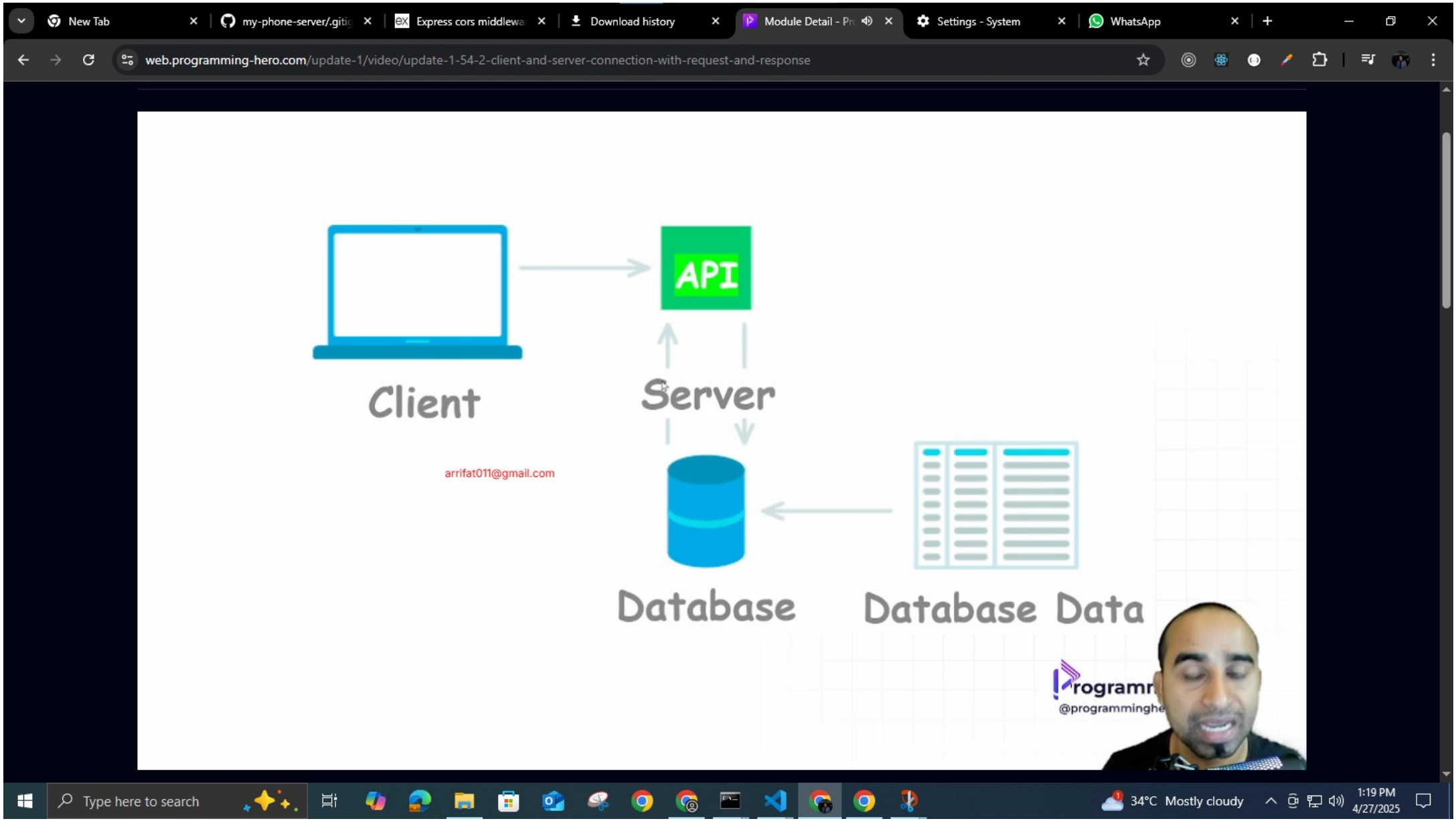


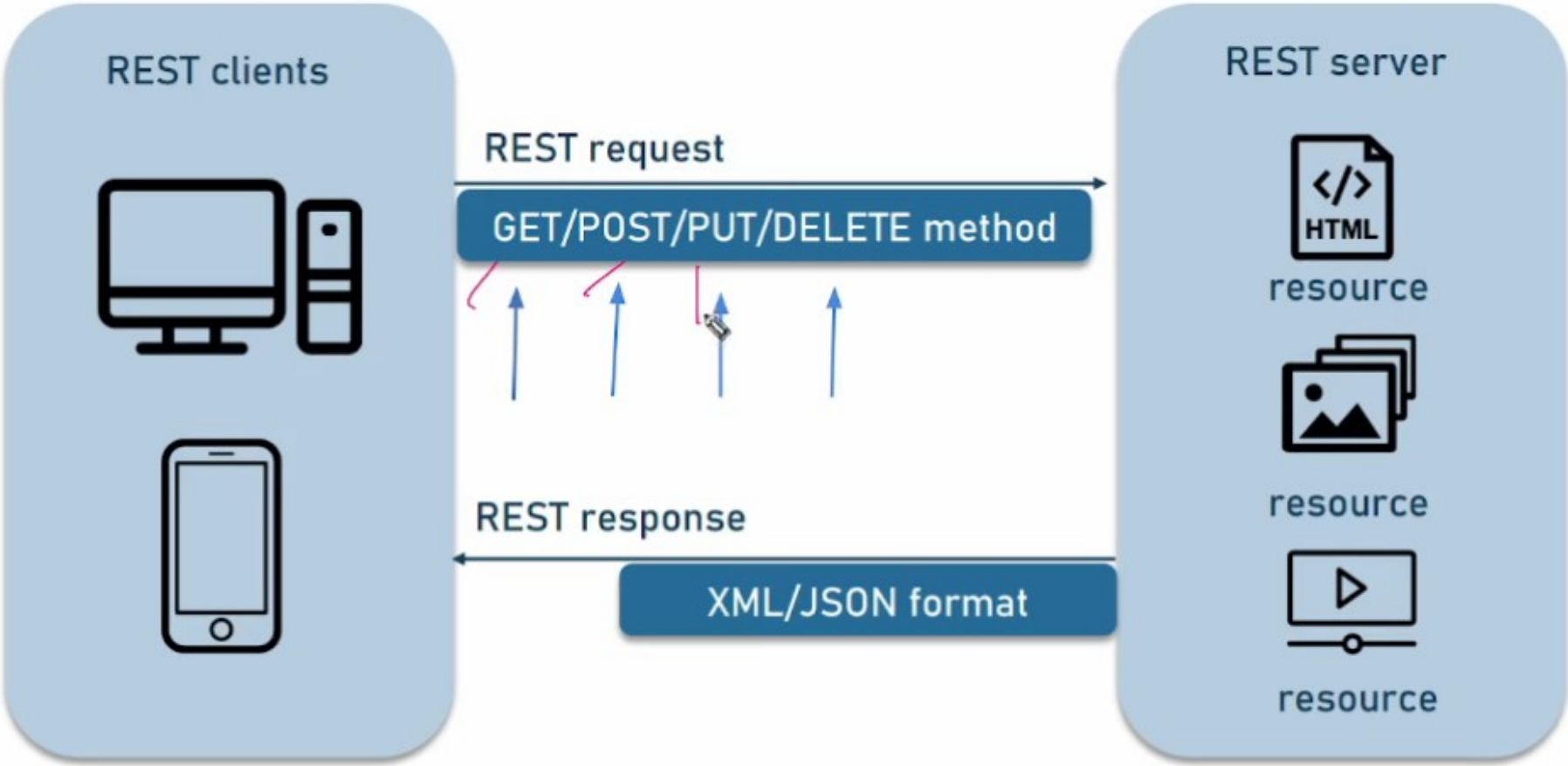

How to
communicate
?????
arrifat011@gmail.com


New Tab
my-phone-server/.gitig
Express cors middlewa
Download history
Module Detail - Pr
Settings - System
WhatsApp
web.programming-hero.com/update-1/video/update-1-54-2-client-and-server-connection-with-request-and-response
Successful
communication
API
arrifat011@gmail.com
@programminghero
2:53 / 12:57
Type here to search
34°C Mostly cloudy
1:18 PM
4/27/2025

তাহলে API ছাড়া কি server & client কে connect করার কোনও way আছে?

-> Noooooo!!!


New Tab
my-phone-server/.gitig
Express cors middlewa
Download history
Module Detail - Pr
Settings - System
WhatsApp
web.programming-hero.com/update-1/video/update-1-54-2-client-and-server-connection-with-request-and-response
API
Client
Server
arrifat011@gmail.com
Database
Database Data
@programminghe
Type here to search
34°C Mostly cloudy
1:19 PM
4/27/2025


REST clients
REST request
GET/POST/PUT/DELETE method
REST response
XML/JSON format
REST server
HTML
resource
resource
resource

Get API =>

সার্ভার এর সাহায্যে ডাটাবেস থেকে ডাটা আনবে

Post API =>

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

Put API =>

সার্ভার এর সাহায্যে ডাটাবেস এর ডাটা আপডেট করবে

rrifat011@gmail.com

Delete API =>

সার্ভার এর সাহায্যে ডাটাবেস এর ডাটা রিমুভ করবে

Let's understand
How POST API connects client & server

বাকি Get, put, delete API
এর জন্য data converstion system একই রকম।

arrifat011@gmail.com

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

```
const coffeeData = { name, chef, supplier,
                     taste, category, details,
                     image, price };

fetch(`http://localhost:5000/coffee`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(coffeeData),
  })
  .then((res) => res.json())
  .then((data) => {
      console.log(data)
  })
```

arrifat011@gmail.com

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

```
const coffeeData = { name, chef, supplier,
                     taste, category, details,
                     image, price };

fetch(`http://localhost:5000/coffee`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(coffeeData),
  })
  .then((res) => res.json())
  .then((data) => {
      console.log(data)
  })
```

```
{
  name: 'Americano Coffee',
  chef: 'Mr. Matin Paul',
  supplier: 'Cappu Authorizer',
  taste: 'Sweet and hot',
  category: 'Americano',
  details: 'Espresso with hot water',
  image: 'https://i.ibb.co/PGqMPr9/11.png',
  price: 890
}
```

arrifat011@gmail.com

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে


```
const coffeeData = { name, chef, supplier,
                     taste, category, details,
                     image, price };

fetch(`http://localhost:5000/coffee`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(coffeeData),
  })
  .then((res) => res.json())
  .then((data) => {
      console.log(data)
  })
```

JSON String

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

```
const coffeeData = { name, chef, supplier,
                     taste, category, details,
                     image, price };

fetch(`http://localhost:5000/coffee`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify(coffeeData),
  })
  .then((res) => res.json())
  .then((data) => {
      console.log(data)
  })
```



```
{
    "name": "Americano Coffee",
    "price": 650,
    "chef": "Mr. Matin Paul",
    "supplier": "Cappu Authorizer",
    "taste": "Sweet and hot",
    "category": "Americano",
    "details": "Espresso with hot water",
    "img": "https://i.ibb.co/PGqMPr9/11.png"
},
```

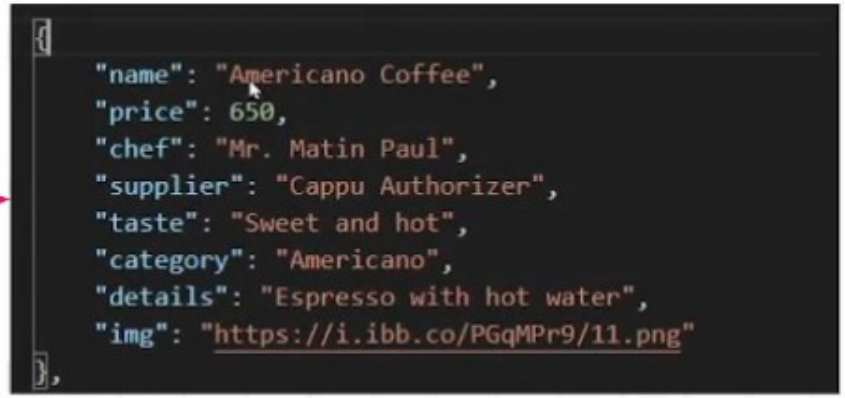
Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

Post API

সার্ভার এর সাহায্যে **ডাটাবেস এ ডাটা রাখবে**

```
app.use(express.json());
```



Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

arrifat011@gmail.com

```
// add a coffee
app.post("/coffee", async (req, res) => {
   const data = req.body;
   const result = await coffeeCollection.insertOne(data);
   res.send(result);
});
```

JavaScript Value

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

```
// add a coffee
app.post("/coffee", async (req, res) => {
   const data = req.body;
   const result = await coffeeCollection.insertOne(data);
   res.send(result);
});
```

```
{
  name: 'Americano Coffee',
  chef: 'Mr. Matin Paul',
  supplier: 'Cappu Authorizer',
  taste: 'Sweet and hot',
  category: 'Americano',
  details: 'Espresso with hot water',
  image: 'https://i.ibb.co/PGqMPr9/11.png',
  price: 890
}
```

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

```
// add a coffee
app.post("/coffee", async (req, res) => {
    const data = req.body;
    const result = await coffeeCollection.insertOne(data);
    res.send(result);
});
```

Response from server
After inserting data
into database

Post API

সার্ভার এর সাহায্যে ডাটাবেস এ ডাটা রাখবে

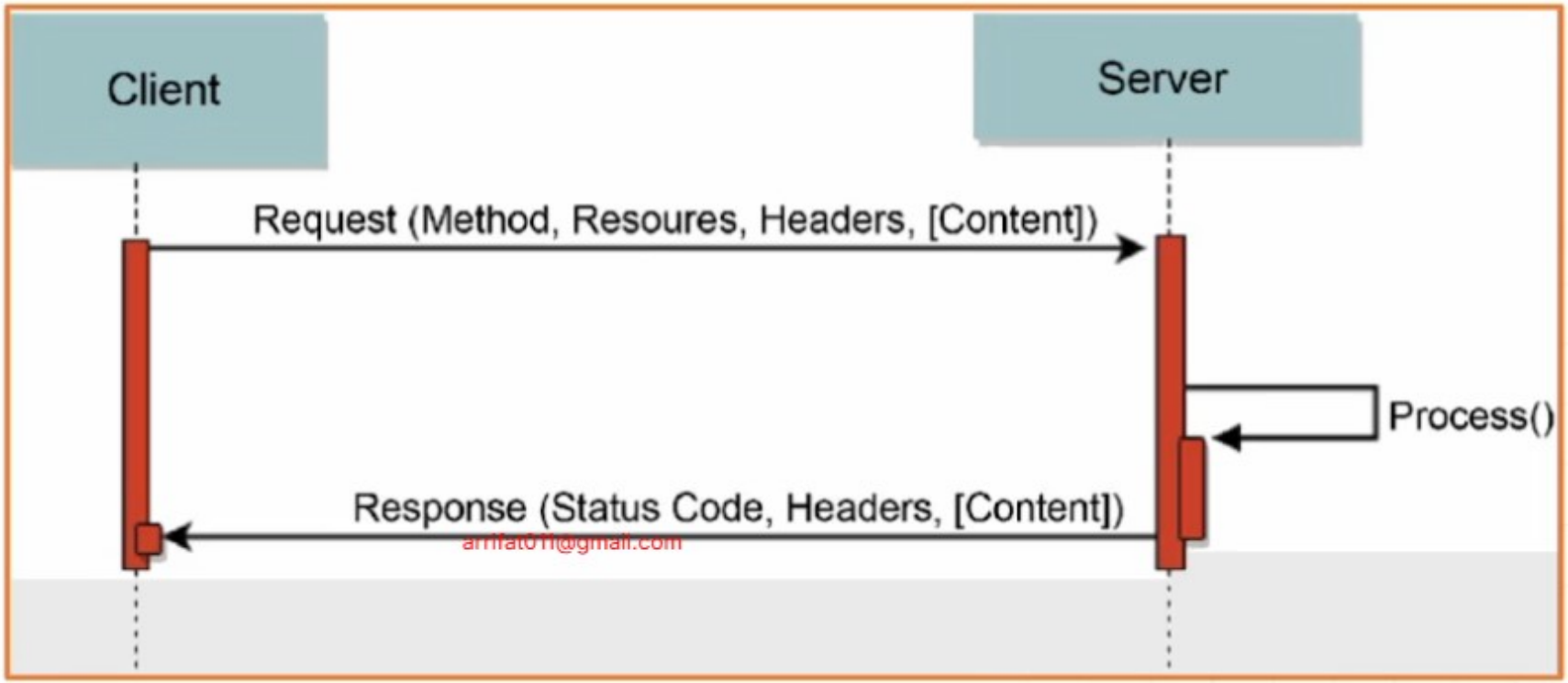
```
// add a coffee
app.post("/coffee", async (req, res) => {
   const data = req.body;
   const result = await coffeeCollection.insertOne(data);
   res.send(result);
});
```

```
{
  acknowledged: true,
  insertedId: new ObjectId("642fdbf802002b9689d82c4c")
}
```

Request
Response

arrifat011@gmail.com

Request/ Response Model



Request (req) object

The req object represents the HTTP request and has properties

- request query string,
- parameters,
- body,
- HTTP headers, and so on

req.body

Contains key-value pairs of data submitted in the request body

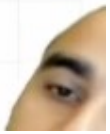
```
app.post('/profile', function (req, res) {
  console.log(req.body)
  res.json(req.body)
})
```

arrifat011@gmail.com

req.params

This property is an object containing properties mapped to the named route "parameters". For example, if you have the route /user/:name, then the "name" property is available as req.params.name. This object defaults to {}.

```
// GET /user/tj
console.dir(req.params.name)
// => 'tj'
```



req.query

This property is an object containing a property for each query string parameter in the route. When query parser is set to disabled, it is an empty object {}, otherwise it is the result of the configured query parser.

Response (res) object

The res object represents the HTTP response that an Express app sends when it gets an HTTP request and has methods

- res.send(),
- res.json(),
- res.status(), res.sendStatus(),
- res.set(), and so on

res.send()

Sends the HTTP response.

The body parameter can be a Buffer object, a String, an object, Boolean, or an Array.

```
res.send({ some: 'json' })
res.send('<p>some html</p>')
```

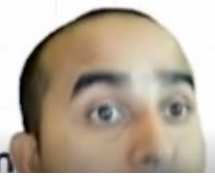
res.json()

arrifat011@gmail.com

Sends a JSON response. This method sends a response (with the correct content-type) that is the parameter converted to a JSON string using JSON.stringify().

The parameter can be any JSON type, including object, array, string, Boolean, number, or null, and you can also use it to convert other values to JSON.

```
res.json(null)
res.json({ user: 'tobi' })
```



res.status()

arrifat011@gmail.com

Sets the HTTP status for the response.

```
res.status(403).end()
res.status(400).send('Bad Request')
res.status(404).sendFile('/absolute/path/to/404.png')
```

res.sendStatus()

Sets the response HTTP status code to statusCode and sends the registered status message as the text response body. If an unknown status code is specified, the response body will just be the code number.

res.set()

Sets the response's HTTP header field to value. To set multiple fields at once, pass an object as the parameter.

```
res.set('Content-Type', 'text/plain')

res.set({
  'Content-Type': 'text/plain',
  'Content-Length': '123',
  ETag: '12345'
})
```

Node JS

arrifat011@gmail.com

Introduction To

Node.js and Package Manager

1995-2009 April

Javascript was created as client side language .
it could only handle the frontend logics

1995-2009 April

- Javascript was created as client side language . it could only handle the frontend logics
- Javascript has no power to communicate with the server.

arrifat011@gmail.com

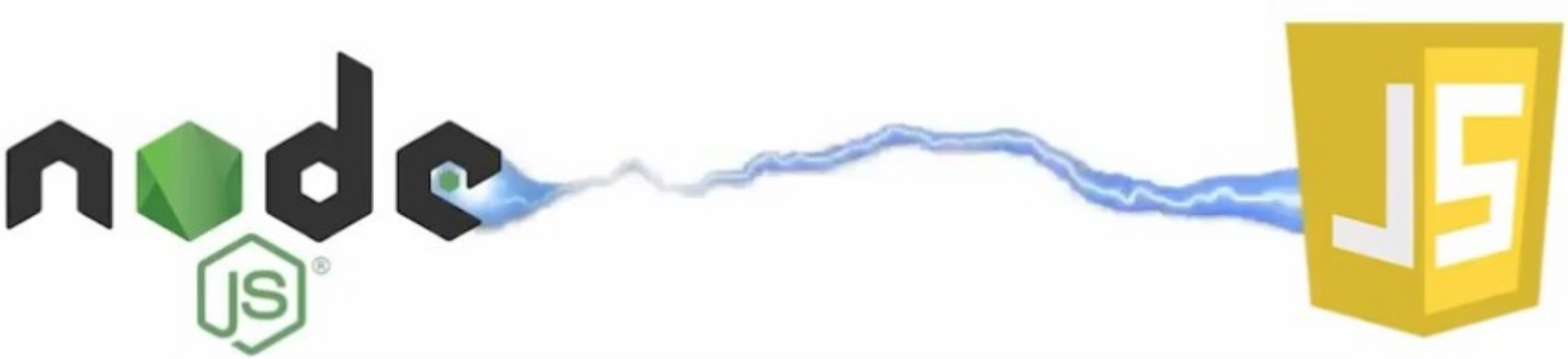
Javascript Could Not do:

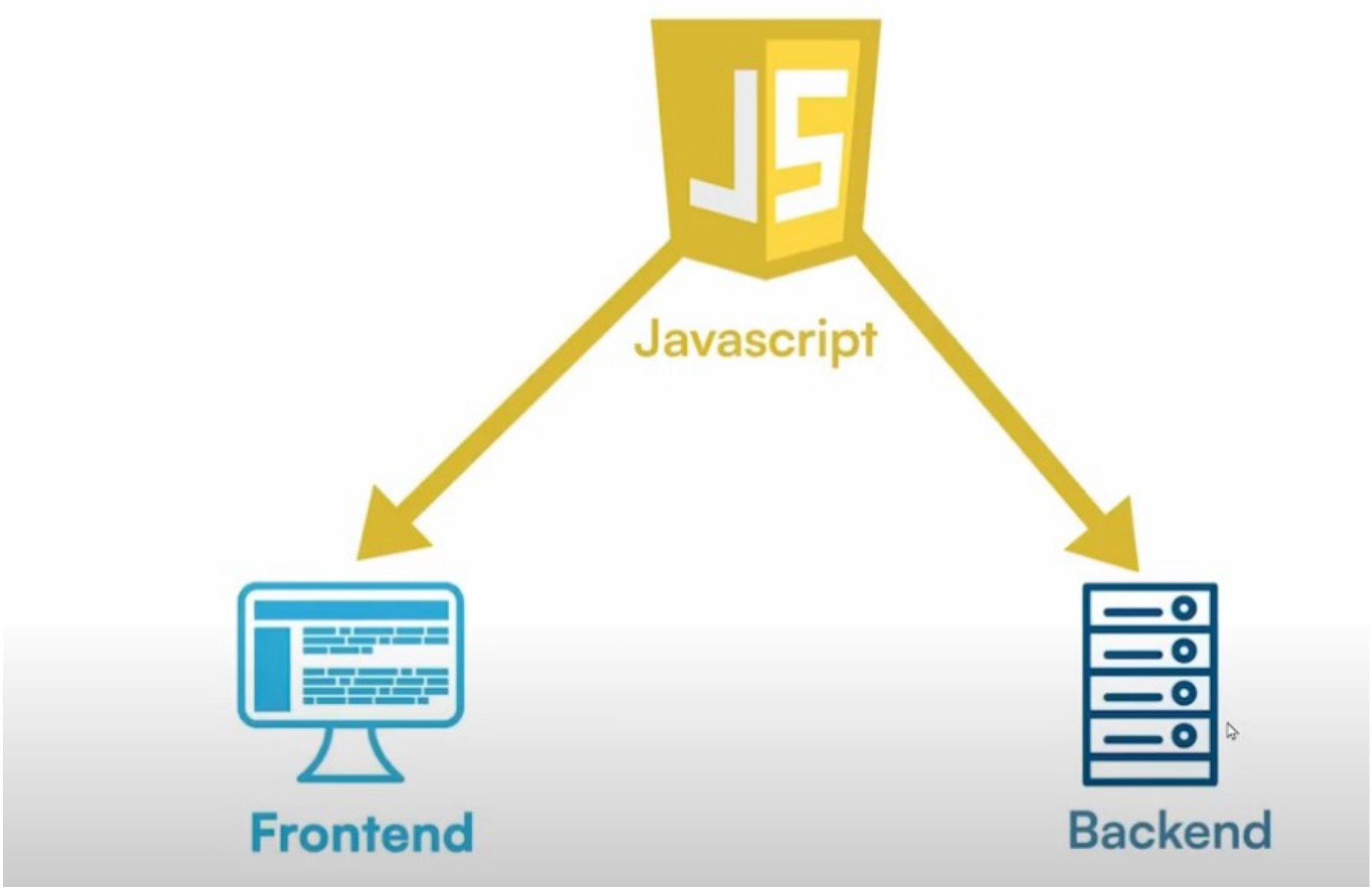
- Query in database
- Handle server side request and response
- Read/Write file on server
- Server side operations

Ryan Dahl

In 2009 May

created a magical thing
which now we are calling
node.js


node
JS


Javascript
Frontend
Backend

Now Javascript can do

- Query in database
- Handle server side request and response
- Read/Write file on server
- Server side operations

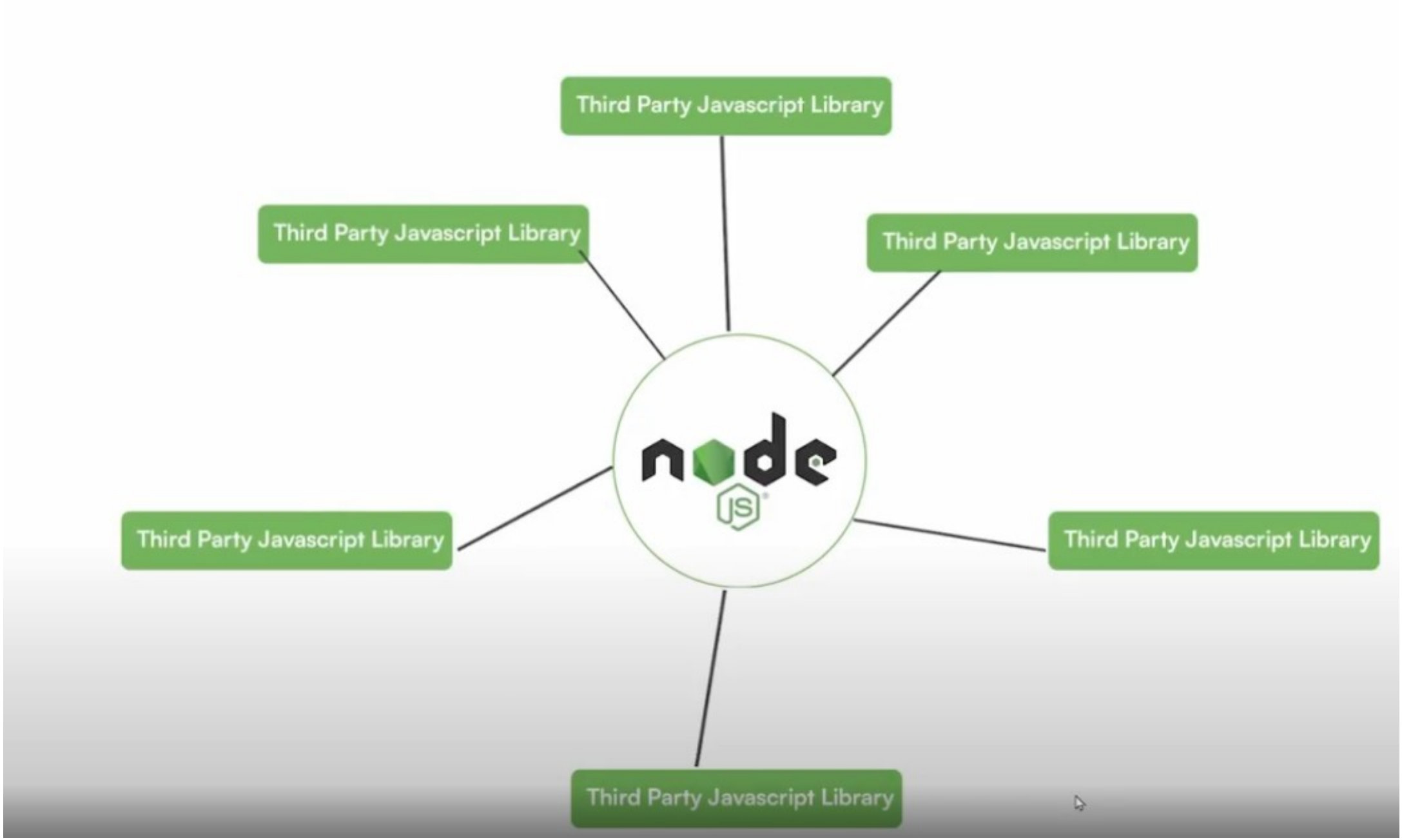
arrifat011@gmail.com

So what is node.js ?

Is a server side javascript runtime

Not a Programming Language ✗

Has Own Library For Manage Node.js


Third Party Javascript Library
Third Party Javascript Library
Third Party Javascript Library
node
JS
Third Party Javascript Library
Third Party Javascript Library
Third Party Javascript Library

Third Party Libraries

What is NodeJS?

Node JS -

- Open source
- Cross platform
- JS runtime environment
- Allows server side scripting
- Single threaded, Non-blocking
- Capable of asynchronous I/O
- Has event-driven architecture

Runtime environment

Different data values and functions are available, and these differences help distinguish front-end applications from back-end applications.

- **Front-end JavaScript** applications are executed in a **browser's runtime** environment and have access to the **window object**.

- **Back-end JavaScript** applications are executed in the **Node runtime** environment and have access to the **file system, databases, and networks attached to the server**.

JS code may be executed in one of **two runtime environments**:

- A browser's runtime environment
- The Node runtime environment