



**LOCUS 2025**  
21st National Technological Festival



# SOFTWARE FELLOWSHIP

Day 7: Integration and Deployment



# </> Prerequisites

- Create a database in MongoDB Atlas

```
mongodb+srv://suravkshrestha:<password>@...
```

- Create a **GitHub** account.
- Create **Vercel** and **Render** accounts, *linking* your **GitHub** account
- Internet connection!

# Part 1

## Integration



# </> How to make API call?



**LOCUS 2025**  
21st National Technological Festival



A X  $\downarrow$  O S

# </> How to make API call?



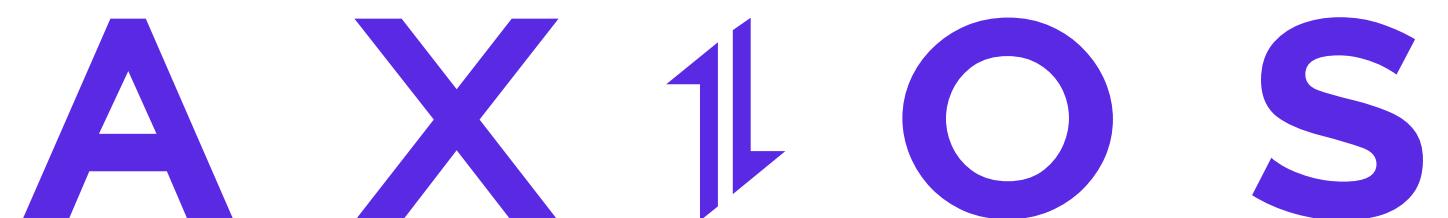
**LOCUS 2025**  
21st National Technological Festival





- Built-in API
- Not all browsers support it

vs



- Need to install via npm/yarn
- Supports all major browsers (even IE)
- Built-in support for many features

If you would like to explore Fetch API:

- [Using the Fetch API \[MDN Docs\]](#)
- [How to Fetch Data from an API Using the Fetch API in JS \[freeCodeCamp\]](#)

# </> Axios request methods

```
axios.get(url, config)
```

```
axios.post(url, data, config)
```

```
axios.delete(url, config)
```

```
axios.patch(url, data, config)
```

- url - The API endpoint to which the request is made.
- data - The data to be sent to the server as the request payload.
- config - Optional configuration settings for the request like headers for authorization, timeout, and other advanced settings.



## frontend/src/components/Todos.jsx

```
useEffect(() => {
  const fetchTodos = async () => {
    try {
      const response = await axios.get(` ${SERVER_URL} /api/v1/todos`);

      setTodos(response.data);
    } catch (error) {
      console.error("Error fetching todos:", error);
    }
  };

  fetchTodos();
}, []);
```

- The frontend and backend servers are hosted on different ports, which leads to CORS (Cross-Origin Resource Sharing) error.

Solution?

Enable CORS in the backend

# </> Create a todo



**LOCUS 2025**  
21st National Technological Festival

```
const handleCreateTodo = async (e) => {
  e.preventDefault();

  try {
    const response = await axios.post(` ${SERVER_URL}/api/v1/todos`, newTodoFormData);

    const newTodo = response.data;

    setTodos([...todos, newTodo]);
  } catch (error) {
    console.error("Error creating todo:", error);
  }

  // Clear form data and close create todo popup
  setNewTodoFormData({ title: "", description: "" });
  setIsCreateTodoActive(false);
};
```

# </> Edit a todo



**LOCUS 2025**  
21st National Technological Festival

```
const handleEditTodo = async (e) => {
  e.preventDefault();

  try {
    const response = await axios.patch(`/${SERVER_URL}/api/v1/todos/${editTodoID}`, editTodoFormData);

    const updatedTodo = response.data;

    const updatedTodos = todos.map((todo) =>
      todo.id === editTodoID ? updatedTodo : todo
    );

    setTodos(updatedTodos);
  } catch (error) {
    console.error("Error editing todo:", error);
  }

  setEditTodoID(null);
  setIsEditTodoActive(false);
};
```

# </> Mark a todo as completed

```
const handleCheckedTodo = async (id) => {
  try {
    const response = await axios.patch(` ${SERVER_URL} /api/v1/todos/${id}`, {
      status: "completed",
    });

    const updatedTodo = response.data;

    const updatedTodoList = todos.map((todo) =>
      todo.id === id ? updatedTodo : todo
    );

    setTodos(updatedTodoList);
  } catch (error) {
    console.error("Error marking todo as completed:", error);
  }
};
```

# </> Delete a todo



```
const handleDeleteTodo = async () => {
  try {
    await axios.delete(` ${SERVER_URL}/api/v1/todos/${deleteTodoID}`);
    const updatedTodoList = todos.map((todo) =>
      todo.id === deleteTodoID ? { ...todo, status: "deleted" } : todo
    );

    setTodos(updatedTodoList);
  } catch (error) {
    console.error("Error deleting todo:", error);
  }

  setDeleteTodoID(null);
  setDeleteTaskConfirmation(false);
};
```

# Part 2

## Deployment





TechAxis

Learn First To Lead The Rest



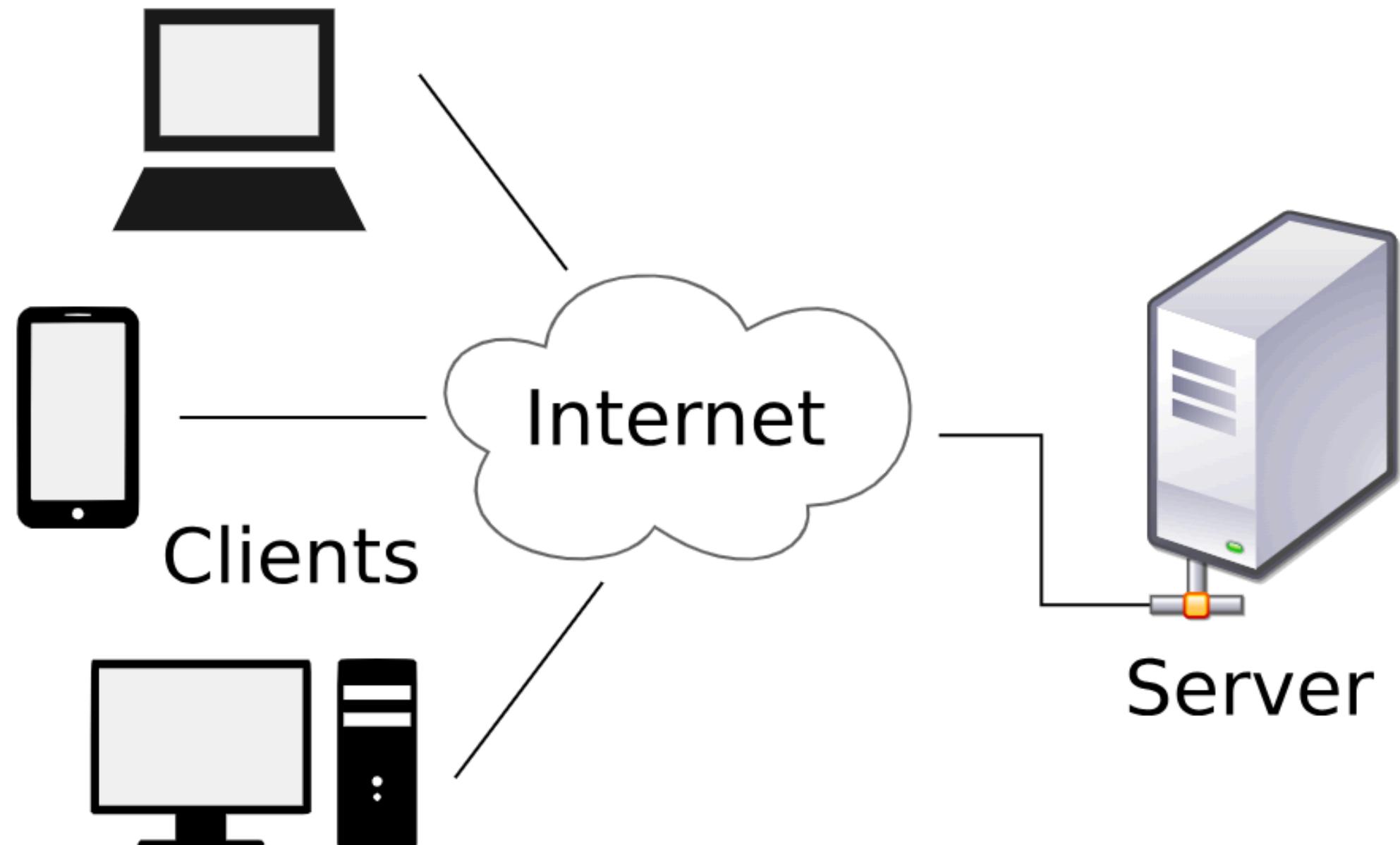
# How does the Internet work?



# </> How does the Internet work?

- **Internet:** Global network of computers
- **Client:** requests services and resources from the server
- **Server:** provides services and resources to other computers (clients) on the network
- **Data center:** Building or room used to house computer servers and related equipment

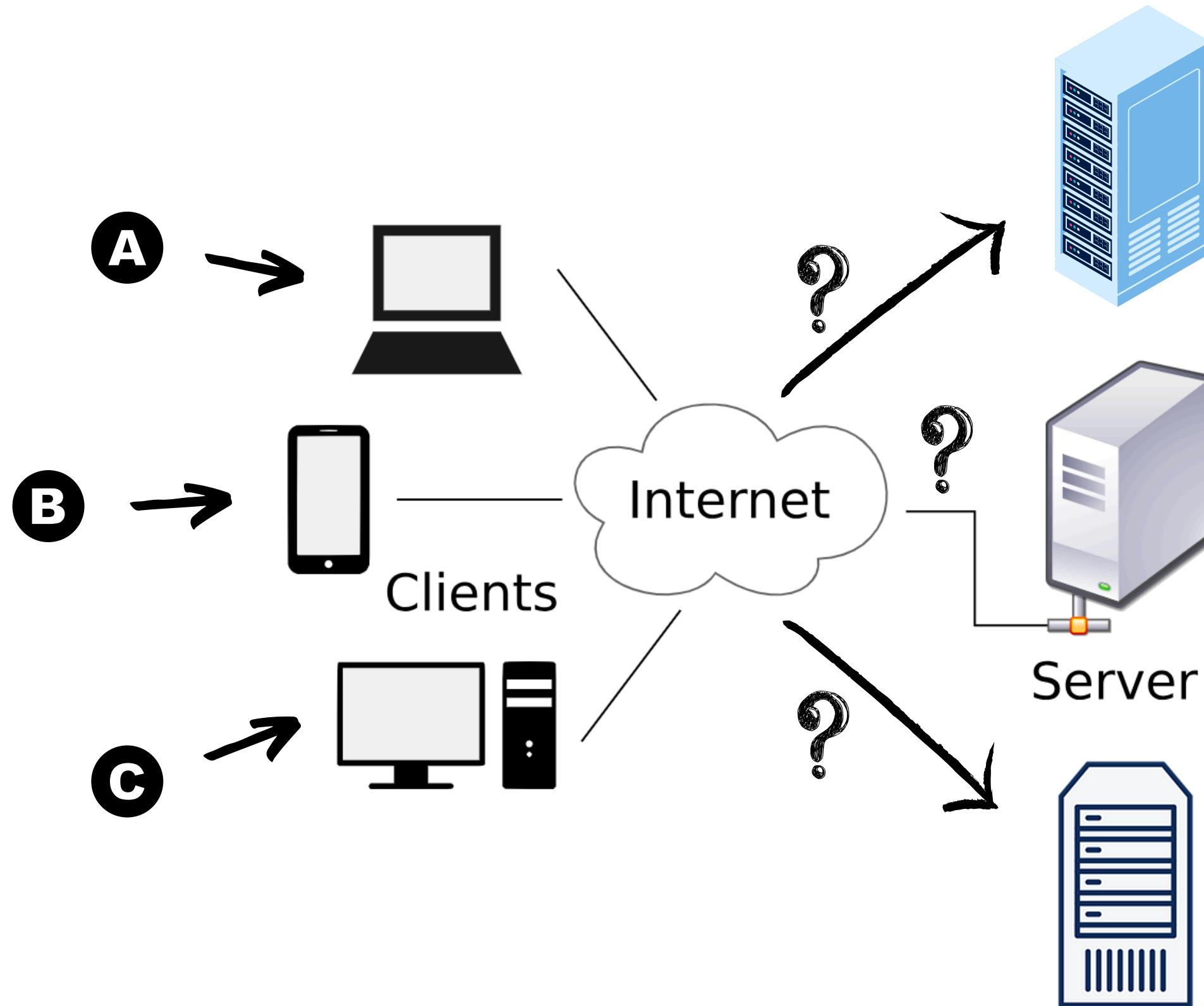
# </> How does the Internet work?



# </> How does the Internet work?



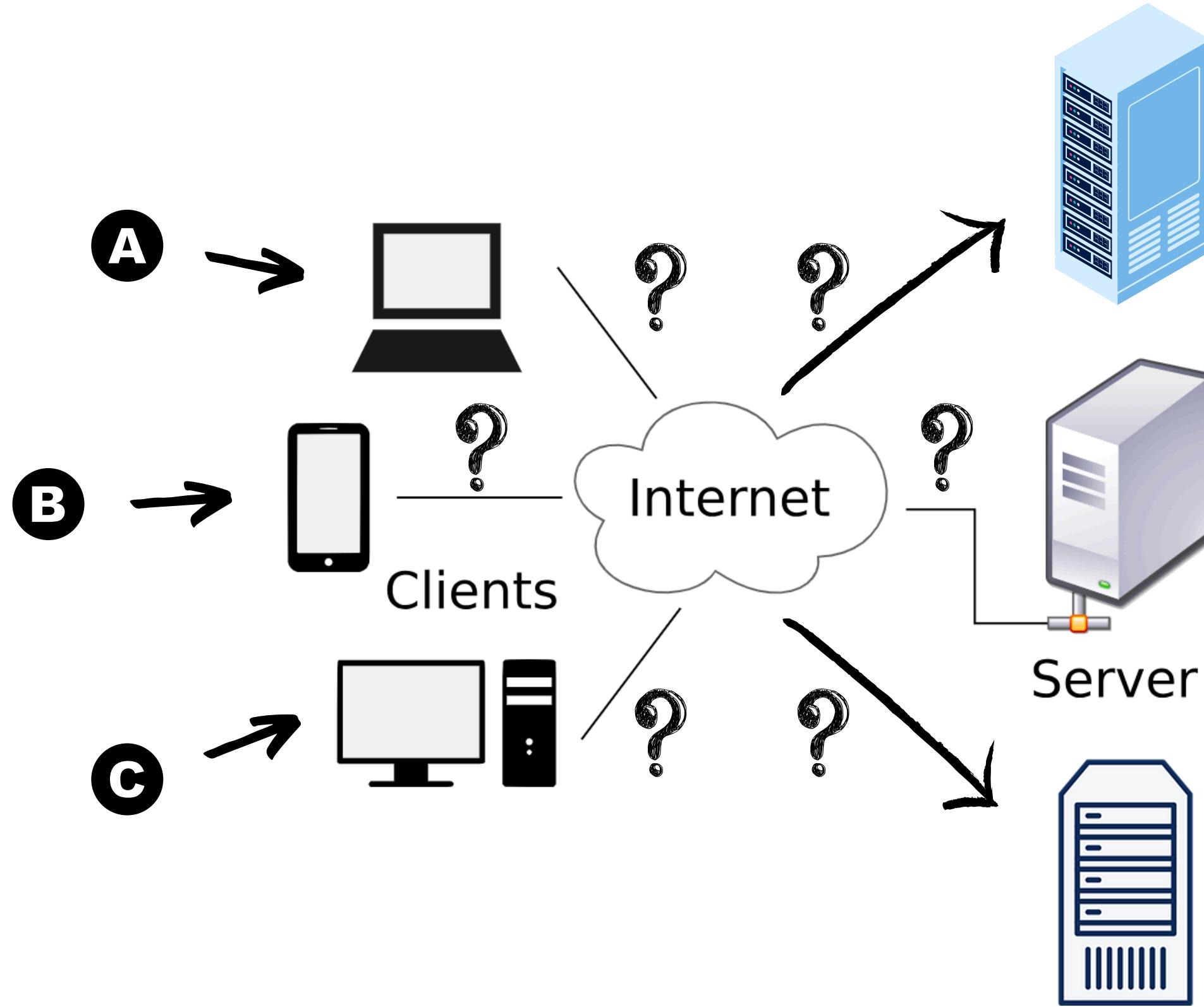
**LOCUS 2025**  
21st National Technological Festival



# </> How does the Internet work?



**LOCUS 2025**  
21st National Technological Festival



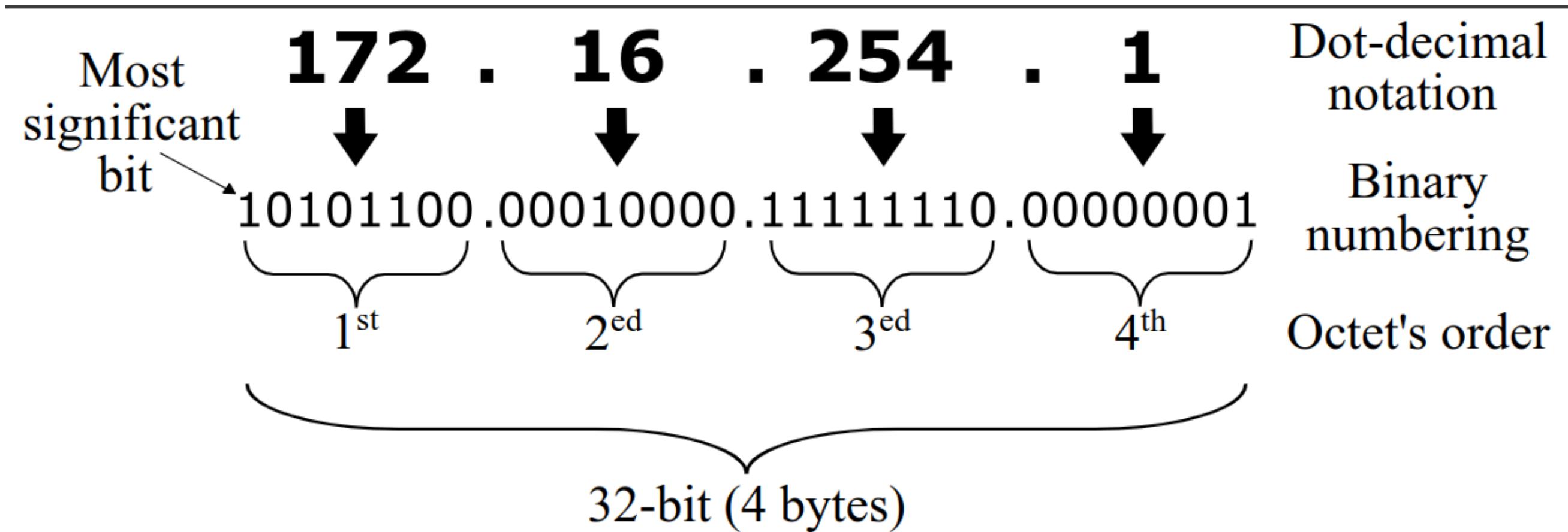
## IP address (IPv4)

- Internet Protocol (IP) address
- unique address assigned to each computer connected to the network

# </> How does the Internet work?



**LOCUS 2025**  
21st National Technological Festival



- Each number range: 0 - 255
- eg. 192.168.10.2, 192.168.1

## Domain name

- human-readable address that maps to an IP address
- example.com, wikipedia.com, locus.com.np

example.com	93.184.215.14
wikipedia.com	103.102.166.226
locus.com.np	172.67.220.23

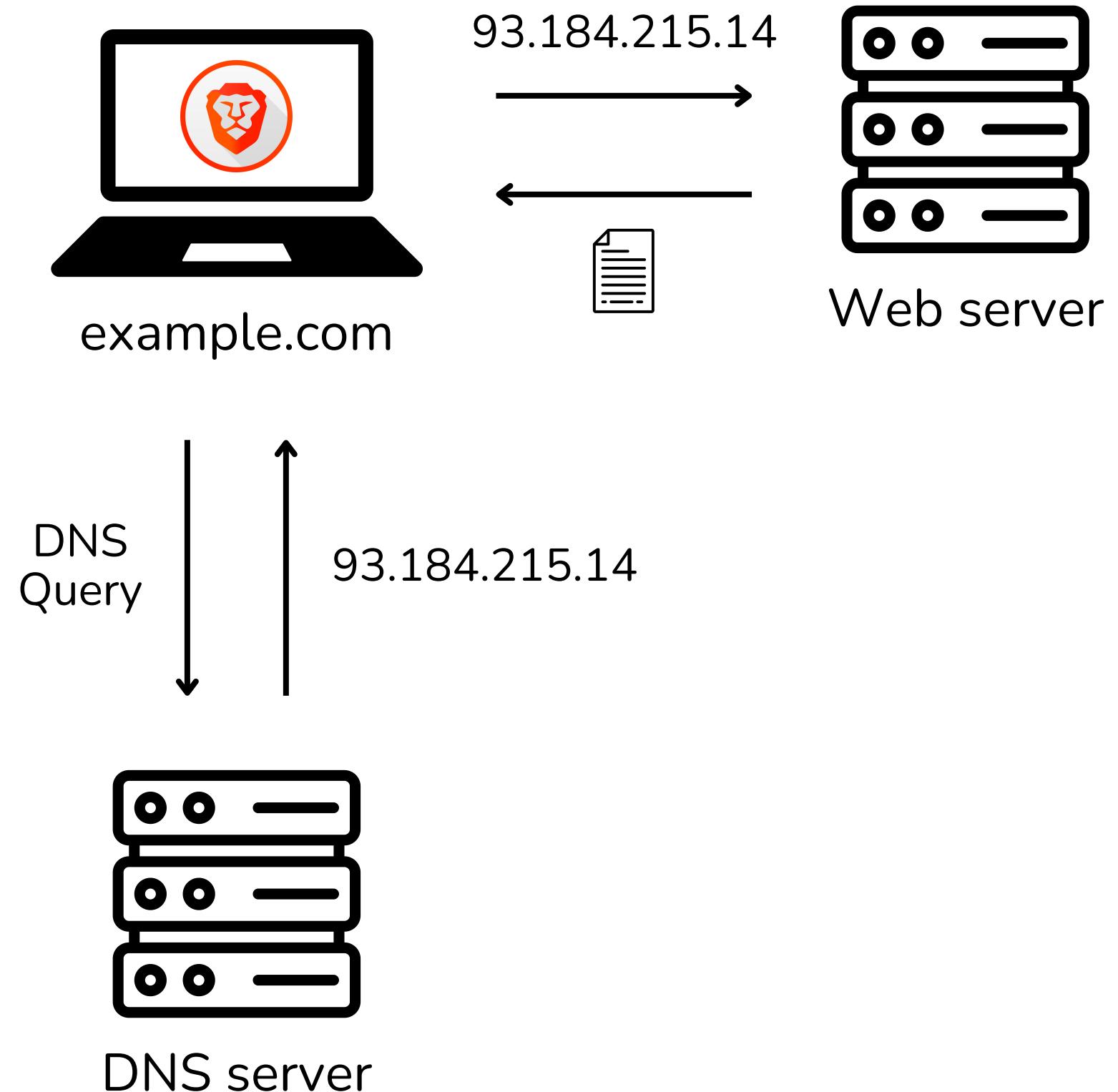
## Domain Name System (DNS)

- translates domain names into IP addresses.
- phonebook of the internet
- server that holds this phonebook is called *DNS server*

# </> How does the Internet work?



LOCUS 2025  
21st National Technological Festival



## Types of Cloud Services

### 1. Infrastructure as a Service (IaaS)

- Provides computing resources over the internet
- eg. Amazon EC2, GCP, Azure, Digital Ocean

## 2. Software as a Service (SaaS)

- Completed product that is run and managed by the service provider
- We do not have to manage anything
- Gmail, Docs, Drive, Dropbox, Zoom

## 3. Platform as a Service (PaaS)

- Provides a platform to build and deploy applications
- eg. *Vercel, Render, Heroku, Railway, Amzon Elastic Beanstalk, Google Cloud Engine (GCP)*

## Frontend

- Vercel
- localhost:5173 -> todos-<rollno>.vercel.app

## Backend

- Render
- localhost:3000 -> todos-<rollno>.onrender.com

## MongoDB Database

- MongoDB Atlas - Database as a Service (DBaaS)



TechAxis

Learn First To Lead The Rest



# Questions?



**JOBAXLE**

A Step & Beyond



TechAxis

Learn First To Lead The Rest



# Quiz Time





# What is the purpose of CORS?

- A. To improve website speed
- B. To manage user authentication
- C. To store cookies
- D. To enable secure cross-origin requests





# Answer:

D





TechAxis  
Learn First To Lead The Rest



# What does DNS stand for?

- A. Domain Network Service
- B. Domain Name System
- C. Data Name Service
- D. Data Network System





TechAxis

Learn First To Lead The Rest



# Answer:

# B



**JOBAXLE**

A Step & Beyond



# What is the purpose of using environment variables in a hosting environment?

- A. To store user data
- B. To manage server performance
- C. To configure application settings securely
- D. To deploy code updates





TechAxis

Learn First To Lead The Rest



# Answer:

# C





# Which platform is used for hosting the API in this fellowship?

- A. Heroku
- B. Vercel
- C. DigitalOcean
- D. Render





# Answer:

D





# How many bits are used in an IPv4 address?

- A. 128
- B. 64
- C. 32
- D. 16





TechAxis

Learn First To Lead The Rest



# Answer:

# C





# Thank you!



Surav Shrestha



suravshrestha

