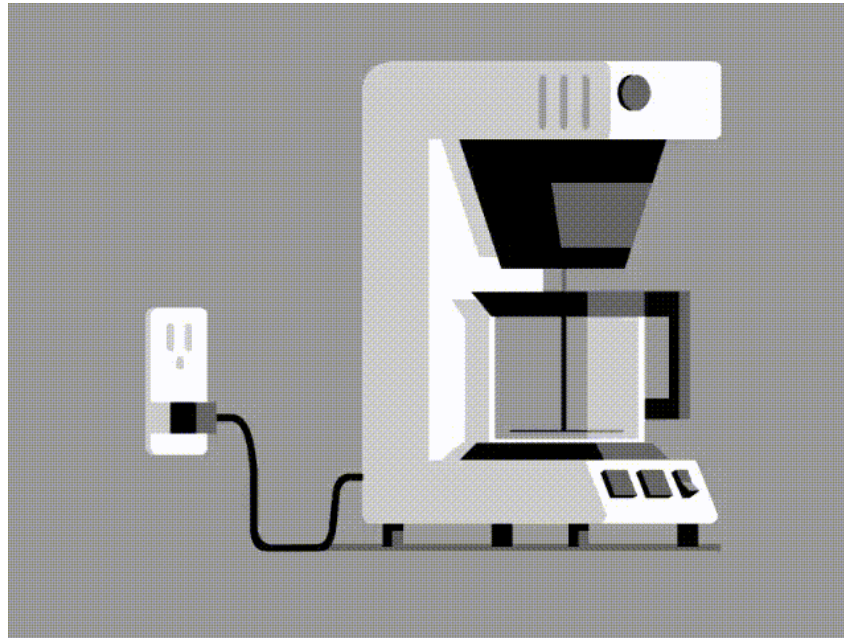# Smart Pour

GROUP 16

E/17/122 - Shazna
E/17/153 - Odasara
E/17/294 - Mishel

Overview

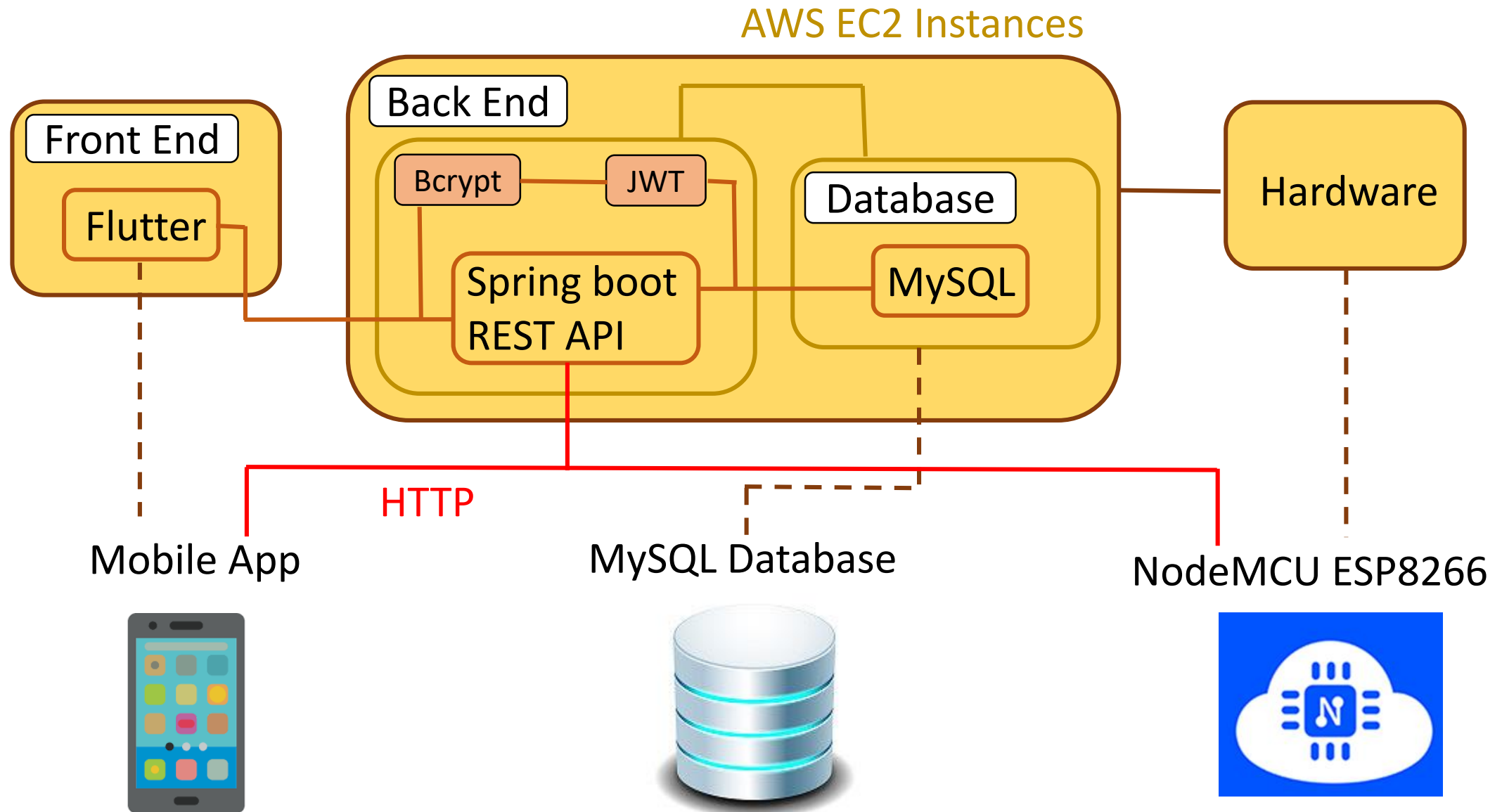# Problem & Solution

Busy Schedules

Long queues in cafeterias

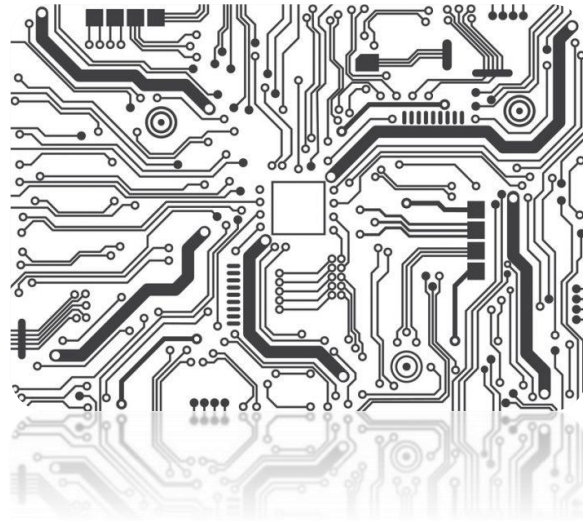Inability to get coffee according to the preference

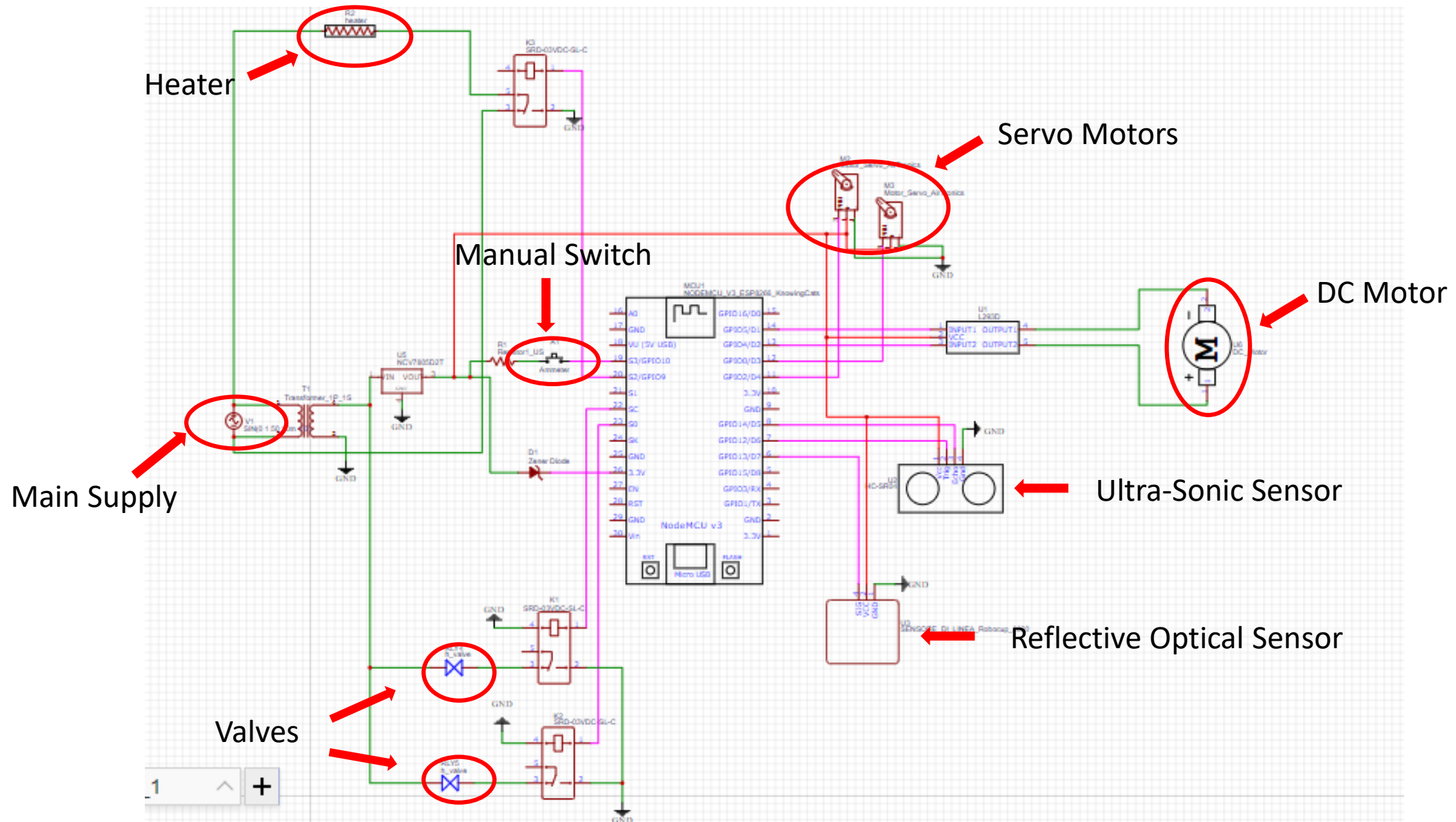An automated coffee machine that can be controlled through a mobile application – **"Smart Pour"**
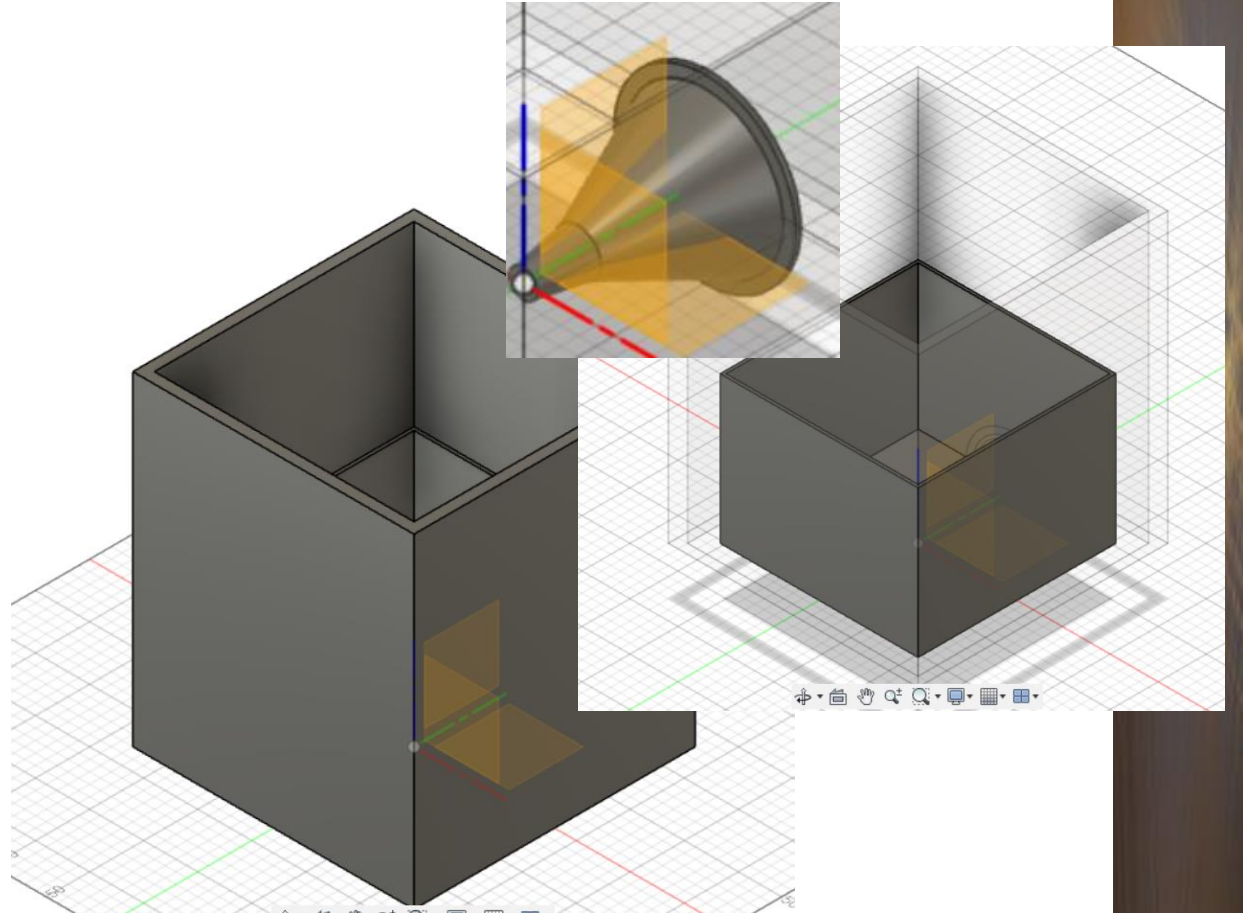
Solution Overview
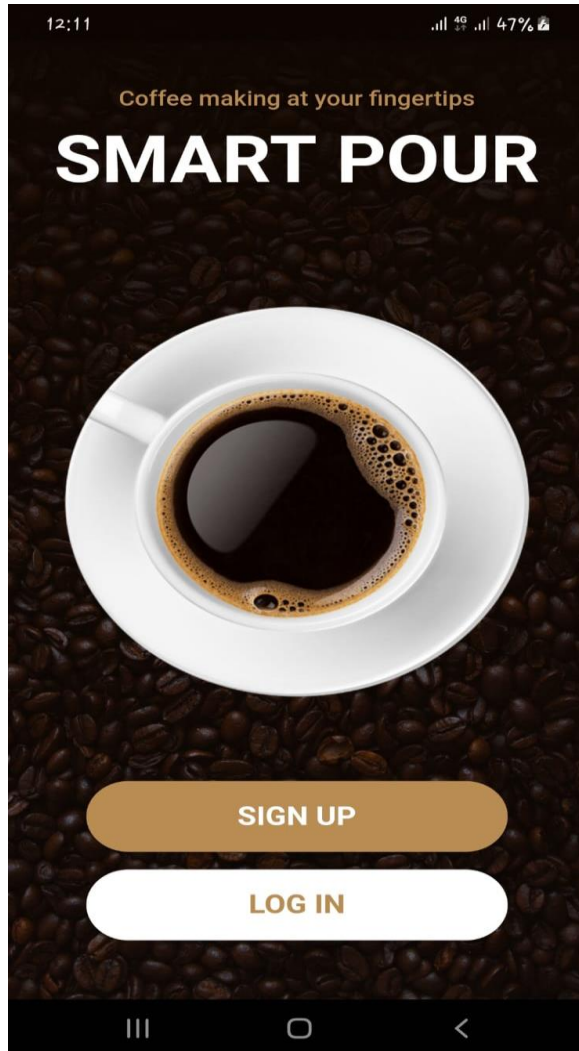
Hardware

Hardware Designs

Hardware Designs

3D Model

CAD Model

Front end

Get logged

Add new devices

Make reservations

Track ingredients

Check schedules

Save favourite recipes

Enhanced User Experience

- Simple design
- Save login credentials
- Dealing with forgotten passwords
- Send notifications
- Easy access for frequently used recipes

Back end

- Deployed complete database and the backend on AWS EC2

=

Instance: i-0f2e668362af9b87c (smart-pour-backend)

| **Details** | Security | Networking | Storage | Status checks | Monitoring | Tags |

▼ Instance summary **Info**

Instance ID
☐ i-0f2e668362af9b87c (smart-pour-backend)

Public IPv4 address
☐ 54.250.249.20 | open address ↗

Private IPv4 addresses
☐ 172.31.0.23

IPv6 address
–

Instance state
⊘ Running

Public IPv4 DNS
☐ ec2-54-250-249-20.ap-northeast-1.compute.amazonaws.com | open address ↗

Hostname type
IP name: ip-172-31-0-23.ap-northeast-1.compute.internal

Private IP DNS name (IPv4 only)
☐ ip-172-31-0-23.ap-northeast-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
☐ 54.250.249.20 [Public IP]

VPC ID
☐ 96fd0e3c6979b6 ↗

AWS Compute Optimizer finding
⊘ Opt into AWS Compute Optimizer for recomm...

Scalability

- The same design can be repeated to make different drinks.

- Use of smaller, independent packages or modules while coding.

- Auto-scaling feature IN Amazon EC2.

- Automatically maintain predictable performance at the lowest possible cost.

Reliability

- Manual mode to operate when there is a network failure.

- Well-secured Features in the back-end and front-end

- Backups of the database provided by AWS

Security

POST ∨ localhost:8080/authenticate

○ form-data   ○ x-www-form-urlencoded   ● raw

```
1 ▾ {
2     "username":"javainuse",
3     "password":"password"
4   }
5
```

Body   Cookies   Headers (11)   Test Results

Pretty   Raw   Preview   JSON ∨ ⇥

```
1 ▾ {
2       "token": "eyJhbGci0iJIUzUxMiJ9.eyJzdWIi0iJqYXZhaW51c2UiLCJleHAi0jE1NTcw0DU5NDAsImlhdCI6MTU1NzA2Nzk0MH0.IOAW8gHFP2Wtnu0
        -4K20z8YiL3w2Q9KT6Ynv0UM2XDU8UtTTdn1geJcA"
```

- **Password Encoding Using Bcrypt**

- **Use of JWT authentication for login and Authorization**

- **The security group in Amazon EC2 acts as a virtual firewall to control incoming and outgoing traffic.**

- **Identity and access management for Amazon EC2.**
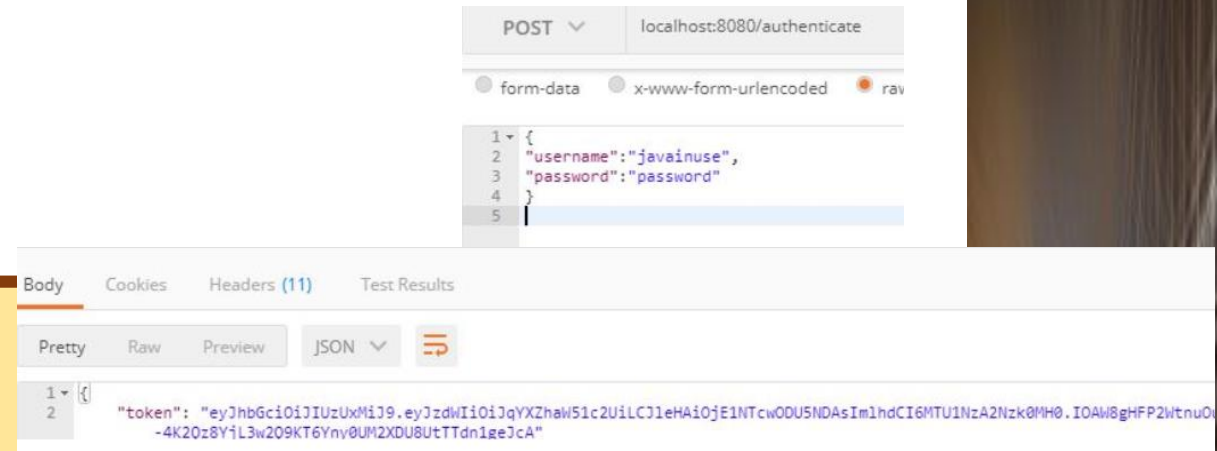
Future Plans

- **Use rechargeable battery power to ensure availability.**
- **Increase the types of beverages.**
- **Increase the amount of servings per request.**
- **Portability - 'Smart Bottle'**

Demonstration