

CORPORATE CLASSIFIEDS

Your personal Online Exchange



OUR TEAM

PRATEEK PUROHIT
(916449)

ASHISH KHETAN
(915605)

MANVENDRA
GUPTA
(915995)

MEGHA AGARWAL
(915927)

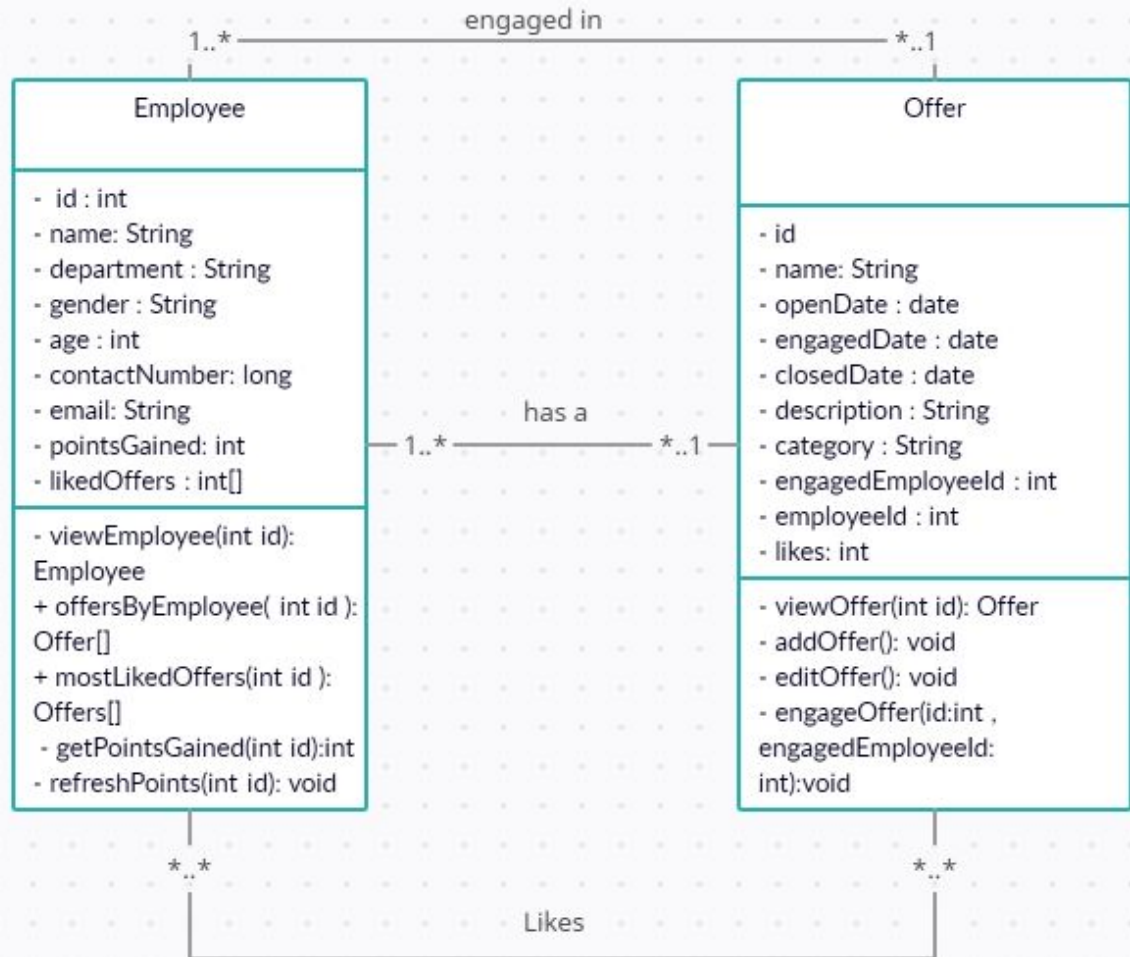
CORPORATE CLASSIFIEDS IN A SHELL



_ REQUIREMENT ANALYSIS



CLASS DIAGRAM



SQL TABLES

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
department	varchar(50)	YES		NULL	
gender	varchar(6)	NO		NULL	
age	int	NO		NULL	
contact_number	bigint	YES		NULL	
email	varchar(100)	YES	UNI	NULL	
points_gained	int	YES		0	

EMPLOYEE
TABLE

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
description	varchar(100)	YES		NULL	
category	varchar(50)	NO		NULL	DEFAULT_GENERATED
open_date	timestamp	YES		CURRENT_TIMESTAMP	
closed_date	timestamp	YES		NULL	
engaged_date	timestamp	YES		NULL	
engaged_emp_id	int	YES	MUL	NULL	
emp_id	int	NO	MUL	NULL	
likes	int	YES		0	

OFFER
TABLE

SQL TABLES (CONTINUED)

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	NULL	
offer_id	int	NO	PRI	NULL	
liked_on	timestamp	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED

LIKED BY TABLE (To describe many to many relationship between offers and employee)

DEVELOPMENT & DESIGNING



— MICROSERVICES

- Auth microservice - to handle authentication of the employee
- Employee microservice - to handle employee related services
- Offer microservice - to handle offer related services
- Points microservice - to refresh points gained by the employee



AUTH _ MICROSERVICE



— REST ENDPOINTS

- **/login** - to log in the employee and generate jwt token
- **/validate** - to validate the jwt token

user-controller User Controller	
POST	/login login
GET	/validate getValidity

A look at swagger.html for auth microservice



Rest Controller (a gist)

```
@PostMapping("/login")
public ResponseEntity<UserToken> login(@RequestBody UserModel user) {
    log.info("Inside Login : ");
    return new ResponseEntity<UserToken>(userServiceImpl.login(user), HttpStatus.OK);
}
```

One of the Post mappings of the rest controller
(to login the user)

**Rest controller calls the
UserServiceImpl.java where all the necessary
details are implemented**

AUTH SERVICE

```
//validates the JWT token
public AuthResponse getValidity(String token) {
    //retrieving the token ( removing the Bearer from the header)
    String token1 = token.substring(7);
    AuthResponse authResponse = new AuthResponse();
    //if valid
    if (jwtUtil.validateToken(token1)) {
        log.info("Token is valid");

        //extract the user name
        String username = jwtUtil.extractUsername(token1);

        //set the values for the response
        authResponse.setUsername(username);
        authResponse.setValid(true);
        authResponse.setEmpid(userRepository.findByEmpUsername(username).getEmpid());
    }
    else {
        authResponse.setValid(false);
        log.error("Token is invalid or expired...");
    }

    return authResponse;
}
```

EXCEPTION HANDLING

- Rest Exception handler - to handle all the exceptions that occurs in the microservice with proper status code
- UnauthorizedException - thrown when the user is invalid.



EMPLOYEE _ MICROSERVICE



— REST ENDPOINTS

- **/employee/viewProfile/{id}** - to view employee details by id parameter
- **/employee/viewEmployeeOffers/{id}** - to view all the offers posted by the employee through id parameter
- **/employee/viewMostLiked/{id}** - to view top 3 most liked offers of the employee by id parameter



Rest Controller (a gist)

```
@CrossOrigin(origins = "http://localhost:4200")
@GetMapping("/viewProfile/{id}")
public ResponseEntity<Employee> viewProfile(@RequestHeader(name = "Authorization", required = true)
    String token,
    @PathVariable("id") int id) throws InvalidUserException, MicroserviceException {
    log.info("Inside view employee profile");
    return new ResponseEntity<>(employeeService.viewEmployee(token, id), HttpStatus.OK);
}
```

One of the Get mappings of the rest controller
(to view the profile of the user)

**Rest controller calls the
employeeServiceImpl.java where all the
necessary details are implemented**

EMPLOYEE SERVICE

```
@Override
public Employee viewEmployee(String token, int id) throws MicroserviceException,
    InvalidUserException {
    log.info("Inside view employee");
    AuthResponse authResponse;
    // authenticate the user
    try {
        authResponse = authClient.getValidity(token).getBody();
    } catch (Exception e) {
        throw new MicroserviceException(e.getMessage());
    }
    // validate the token
    if (authResponse.isValid()) {
        if (authResponse.getEmpid() != id) {
            throw new InvalidUserException("token is invalid for the given user");
        }
        // retrieve the employee details
        Employee employee = employeeRepository.findById(id).orElse(null);
        if (employee == null) {
            log.error("Invalid employee id");
            throw new NoSuchElementException();
        }
        return employee;
    } else {
        log.error("Token invalid");
        throw new InvalidUserException("Invalid User");
    }
}
```

— EXCEPTION HANDLING

- Employee Exception handler - to handle all the exceptions that occurs in the microservice with proper status code
- InvalidUserException - thrown when the user is unauthorized
- MicroserviceException - thrown when there is an error in the communicated microservice.



FEIGN CLIENTS

- Auth client - to communicate with the authentication microservice
- Offer client - to communicate with the offer microservice

```
//to connect to authentication service
@FeignClient(url = "${auth.feign.client}", name = "${auth.feign.name}")
public interface AuthClient {

    // checks the validity of the jwt token
    @RequestMapping(path = "/validate", method = RequestMethod.GET)
    public ResponseEntity<AuthResponse> getValidity(
        @RequestHeader(name = "Authorization", required = true) String token);
}
```

OFFER _ MICROSERVICE



— REST ENDPOINTS

- **/offer/addOffer** - to post a new offer
- **/offer/editOffer** - to edit an existing offer
- **/offer/engageOffer** - to engage an offer
- **/offer/getOfferByCategory** - to filter offers by category
- **/offer/getOfferByPostedDate** - to filter offers by posted date
- **/offer/getOfferByTopLikes** - to retrieve top 3 most liked offers
- **/offer/getOfferDetails/{id}** - to retrieve offer by id parameter



Rest Controller (a gist)

```
@CrossOrigin(origins = "http://localhost:4200")
@GetMapping("/getOfferDetails/{offerId}")
public Offer getOfferDetails(@RequestHeader("Authorization") String token,
    @PathVariable("offerId") int offerId)
    throws OfferNotFoundException, InvalidTokenException, MicroserviceException {

    log.debug("inside getOfferDetails method of offer microservice");
    Offer offer = offerService.getOfferDetails(token, offerId);
    return offer;
}
```

One of the Get mappings of the rest controller (to get the details of an offer by offer id)

Rest controller calls the offerService.java where all the necessary details are implemented

OFFER SERVICE

```
public Offer getOfferDetails(String token, int offerId)
    throws OfferNotFoundException, InvalidTokenException,
        MicroserviceException {
    // authenticate the user
    ResponseEntity<AuthResponse> response;
    try {
        response = authClient.verifyToken(token);
    } catch (Exception e) {
        log.info("some error in auth microservice");
        throw new MicroserviceException(e.getMessage());
    }
    // check if token is valid
    if (response.getBody().isValid()) {
        Optional<Offer> offer = offerRepository.findById(offerId);
        // if offer is not found
        if (!offer.isPresent())
            throw new OfferNotFoundException("No offer found");

        return offer.get();
    }
    // if token is invalid
    else {
        throw new InvalidTokenException("token is invalid");
    }
}
```


EXCEPTION HANDLING

- **Global Exception handler** - to handle all the exceptions that occurs in the microservice with proper status code
- **EmployeeNotFoundException** - thrown when no employee is found
- **ImproperDateException** - thrown when improper date is passed
- **InvalidTokenException** - thrown when the jwt token is invalid
- **MicroserviceException** - thrown when the communicated microservice is not working
- **OfferNotFoundException** - thrown when no offers are found



FEIGN CLIENTS

- Auth client - to communicate with the authentication microservice
- Employee client - to communicate with the employee microservice

```
//to connect to authentication service
@FeignClient(url = "${auth.feign.client}", name = "${auth.feign.name}")
public interface AuthClient {

    // checks the validity of the jwt token
    @RequestMapping(path = "/validate", method = RequestMethod.GET)
    public ResponseEntity<AuthResponse> getValidity(
        @RequestHeader(name = "Authorization", required = true) String token);
}
```

POINTS _ MICROSERVICE



— REST ENDPOINTS

- **/getpointsbyemp/{id}** - to get points gained by the employee
- **/refreshpointsbyemp/{id}** - to refresh the points of the employee

points-controller Points Controller

GET

/getpointsbyemp/{id} getPointsByEmpId

POST

/refreshpointsbyemp/{id} refreshPointsByEmpId

A look at swagger (for points controller)



Rest Controller (a gist)

```
@GetMapping("/getpointsbyemp/{id}")  
public ResponseEntity<Integer> getPointsByEmpId(@RequestHeader(name = "Authorization")  
    String token,  
    @PathVariable("id") int id) throws MicroserviceException {  
    log.info("Inside getpointsbyemployeeid of points microservice");  
    return new ResponseEntity<>(pointsService.getPoints(token, id), HttpStatus.OK);  
}
```

One of the Get mappings of the rest controller (to get the points gained by the employee)

Rest controller calls the pointsServiceImpl.java where all the necessary details are implemented

POINTS SERVICE

```
public Integer getPoints(String token, int id) throws MicroserviceException,
    InvalidUserException {
    log.info("Inside getpoints");
    AuthResponse authResponse;
    // verify the token
    try {
        authResponse = authClient.verifyToken(token).getBody();
    } catch (Exception e) {
        throw new MicroserviceException(e.getMessage());
    }
    // validate the user
    if (authResponse.isValid()) {
        Integer points;
        // retrieve the points
        try {
            points = offerClient.getPointsById(token, id);
        } catch (Exception e) {
            throw new MicroserviceException(e.getMessage());
        }
        return points;
    } else {
        log.error("Token invalid");
        throw new InvalidUserException("Invalid User");
    }
}
```

EXCEPTION HANDLING

- **InvalidUserException** - thrown when the jwt token is invalid
- **MicroserviceException** - thrown when the communicated microservice is not working
- **PointsExceptionHandler** - to handle all the exceptions thrown by the rest controller



FEIGN CLIENTS

- Auth client - to communicate with the authentication microservice
- Employee client - to communicate with the employee microservice
- Offer client - to communicate with the offer microservice

```
@FeignClient(url = "${offer.feign.client}", name = "${offer.feign.name}")
public interface OfferClient {

    @GetMapping("/getOffers/{emp_id}")
    public List<Offer> getOfferByEmpId(@RequestHeader(name = "Authorization")
        String token,
        @PathVariable("emp_id") int id);

    @GetMapping("/getPoints/{emp_id}")
    public int getPointsById(@RequestHeader("Authorization")
        String token,
        @PathVariable("emp_id") int id);
}
```

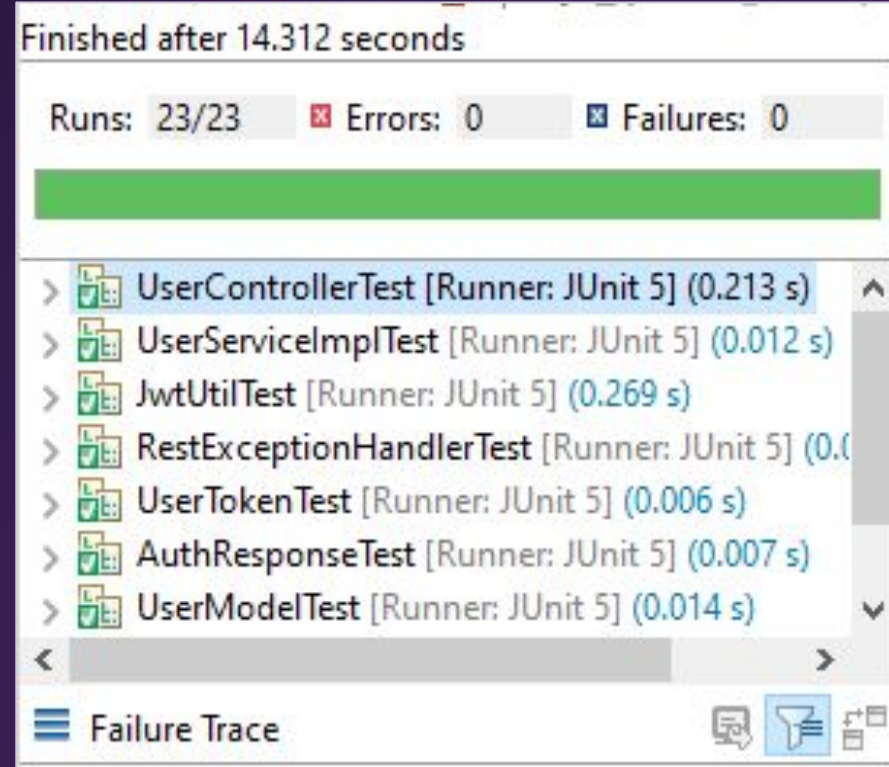

__ TESTING



— AUTH MICROSERVICE

- AuthMicroserviceApplicationTests
- UserControllerTest
- RestExceptionHandlerTest
- JwtUtilTest
- ModelTests
- UserServiceImplTest

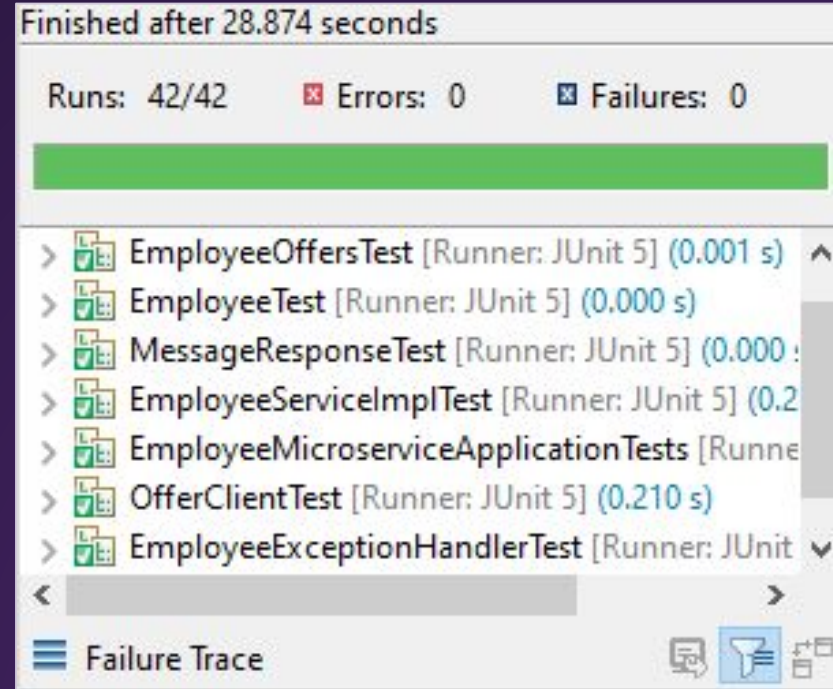
There are 23 test cases and all of them pass successfully



— EMPLOYEE MICROSERVICE

- EmployeeMicroserviceApplicationTests
- OfferClientTest
- EmployeeControllerTest
- EmployeeExceptionHandlerTest
- ModelTests
- EmployeeServiceImplTest

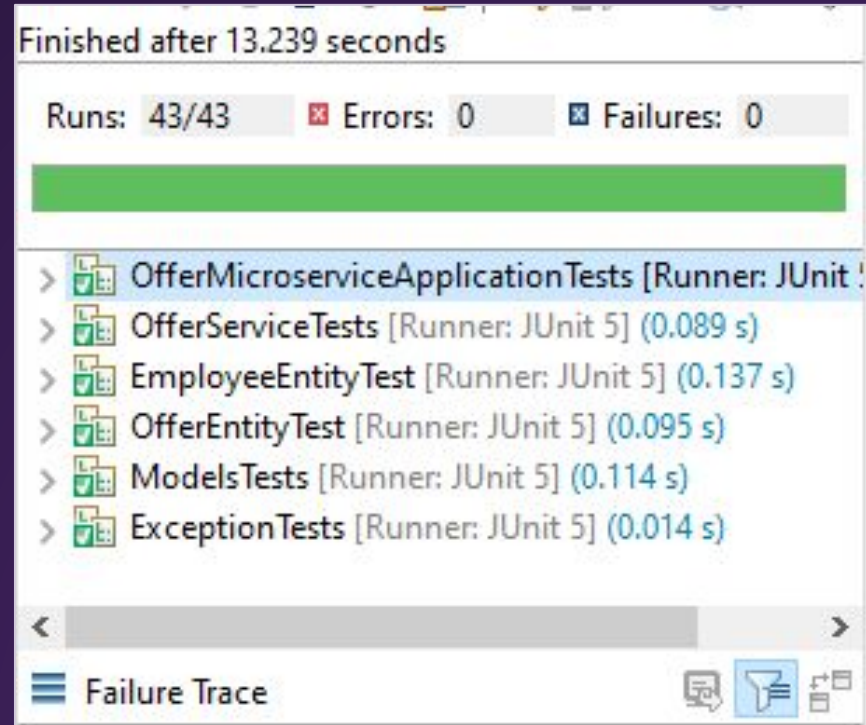
There are 42 test cases and all of them pass successfully



— OFFER MICROSERVICE

- EmployeeEntityTests
- ExceptionTests
- OfferEntityTests
- OfferServiceTests
- ModelTests

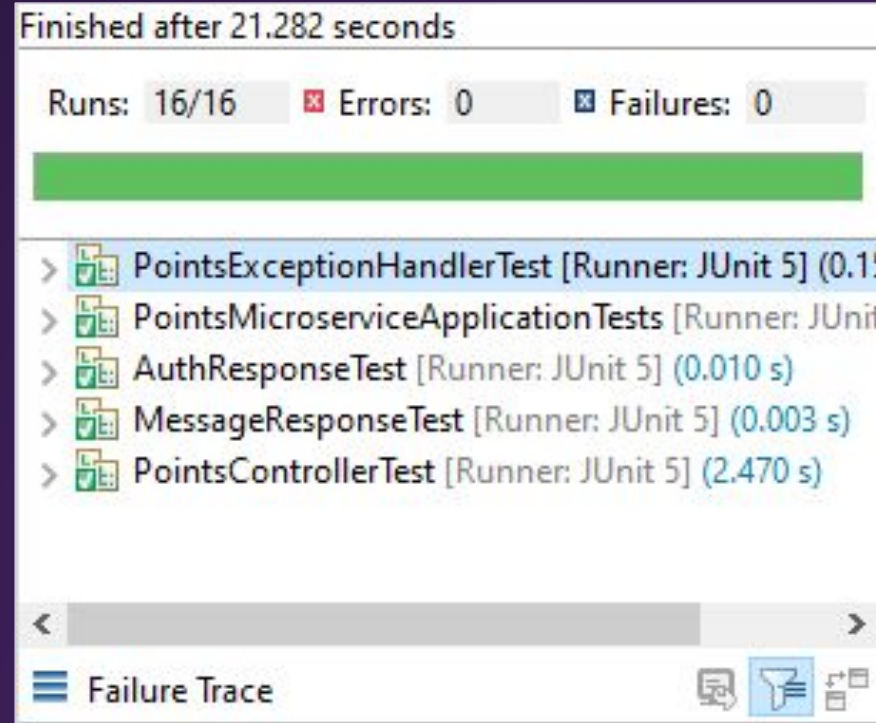
There are 43 test cases and all of them pass successfully



POINTS MICROSERVICE

- PointsMicroserviceApplicationTest
- PointsControllerTests
- PointsExceptionHandlerTests
- ModelTests

There are 16 test cases and all of them pass successfully



USER INTERFACE





Welcome to Corporate Classifieds

Your personal Online Exchange

[Login](#)

LOGIN PAGE

Login

Username

Password

Sign in

Login

Username

Username must be at least 3 characters long.

Password

Password is required.

Sign in

Proper validation for each field

MAIN PAGE

Options to filter data by category,
top likes or by posted date



Corporate Classifieds

Home

Contact us

Logout



Filter by category

Filter by top likes

filter by posted-date

search

My Profile

Post an offer

My offers

view recently liked offers

offer changed on 30th july

Posted on: July 23, 2021

Engaged: not yet

Status: open

Category: Electronics

Likes: 1

Open

Offer new for 29th july

Posted on: July 23, 2021

Engaged: not yet

Status: open

Category: Electronics

Likes: 1

Open

offer 3

Posted on: July 23, 2021

Engaged: engaged

Status: open

Category: electronics

Likes: 2

Open

PROFILE PAGE

Profile

Properties	Value
Id	2
Name	mr faisu
Department	not available
Gender	other
Age	31
Contact Number	not available
Email	not available
Points Gained	300 

[Back](#)

POST OFFER PAGE

Post an Offer

Properties

Value

Name

name of the offer is required.

Description

Description is required.

Category

Submit

Go back

MY OFFERS PAGE

My Offers

offer changed on 30th july

Posted on: July 23, 2021

Engaged: not yet

Status: open

Category: Electronics

Likes: 1

Open

Edit

Offer new for 29th july

Posted on: July 23, 2021

Engaged: not yet

Status: open

Category: Electronics

Likes: 1

Open

Edit

offer last

Posted on: July 26, 2021

Engaged: not yet

Status: open

Category: electro

Likes: 0

Open

Edit

EDIT OFFER PAGE

Offer details

Properties	Value
Id	3
Name	<input type="text" value="offer changed on 30th july"/>
Description	<input type="text" value="this is a new description for offer 3 updated"/>
Category	<input type="text" value="Electronics"/>
Posted on	July 23, 2021
Engaged on	not yet
Status	open
Likes	1

[Submit](#)[Go back](#)

OFFER DETAILS PAGE

Offer details

Properties	Value
Id	3
Name	offer changed on 30th july
Description	this is a new description for offer 3 updated
Category	Electronics
Posted on	July 23, 2021
Engaged on	not yet
Status	open
Likes	1

Liked the post?



Go back

Engage

Contact Details

Your problems are our priority

Services	Values
Phone	+9971930070
email	classifieds@gmail.com
Address	Noida Yamuna Expressway Road, Green Park, Sector 135, Noida, Uttar Pradesh 201304

[Back](#)

— DEPLOYMENT



AUTH MICROSERVICE

✓ Build Succeeded

Pipeline execution ID: [3c1a7a66-e925-44c6-83eb-ab575686f605](#)

Build



[AWS CodeBuild](#)

✓ Succeeded - 13 minutes ago

[Details](#)

[3aaa6d91](#) [Source: port changed](#)

Build Stage

✓ Deploy Succeeded

Pipeline execution ID: [3c1a7a66-e925-44c6-83eb-ab575686f605](#)

Deploy



[AWS Elastic Beanstalk](#) [↗](#)

✓ Succeeded - 13 minutes ago

[3aaa6d91](#) [Source: port changed](#)

Deploy Stage

EMPLOYEE MICROSERVICE

✔ **Build** Succeeded

Pipeline execution ID: [652f380c-65fb-45ed-a04d-ae8f52574d7a](#)

Build

[AWS CodeBuild](#)

✔ **Succeeded** - 29 minutes ago

[Details](#)

[033ce814](#) [🔗](#) Source: sixth commit

Build Stage

✔ **Source** Succeeded

Pipeline execution ID: [652f380c-65fb-45ed-a04d-ae8f52574d7a](#)

Source

[GitHub \(Version 2\)](#) [🔗](#)

✔ **Succeeded** - 33 minutes ago

[033ce814](#) [🔗](#)

[033ce814](#) [🔗](#) Source: sixth commit

Source Stage

OFFER MICROSERVICE

✔ Build Succeeded

Pipeline execution ID: [adddbe2c-bd58-4ccf-993b-4f6654264a88](#)

Build



[AWS CodeBuild](#)

✔ Succeeded - 3 hours ago

[Details](#)

[20d3ca74](#) [🔗](#) Source: port changed

Build Stage

✔ Source Succeeded

Pipeline execution ID: [adddbe2c-bd58-4ccf-993b-4f6654264a88](#)

Source



[GitHub \(Version 2\)](#) [🔗](#)

✔ Succeeded - 3 hours ago

[20d3ca74](#) [🔗](#)

[20d3ca74](#) [🔗](#) Source: port changed

Source Stage

POINTS MICROSERVICE

✔ Build Succeeded

Pipeline execution ID: [9a153d8a-3d8c-4e07-8a77-f7bdb276e0aa](#)

Build

[AWS CodeBuild](#)

✔ Succeeded - 8 minutes ago

[Details](#)

[cabf6d3d](#) [🔗](#) Source: points microservice added

Build Stage

✔ Deploy Succeeded

Pipeline execution ID: [9a153d8a-3d8c-4e07-8a77-f7bdb276e0aa](#)

Deploy

[AWS Elastic Beanstalk](#) [🔗](#)

✔ Succeeded - 7 minutes ago

[cabf6d3d](#) [🔗](#) Source: points microservice added

Deploy Stage

— **THANK YOU**

