

TASK

Code

```
package com.bajajfinserv.hiring;

import com.fasterxml.jackson.annotation.JsonProperty;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.http.*;
import org.springframework.web.client.RestTemplate;

import java.util.HashMap;
import java.util.Map;

@SpringBootApplication
public class HiringApplication implements CommandLineRunner {

    public static void main(String[] args) {
        SpringApplication.run(HiringApplication.class, args);
    }

    @Override
    public void run(String... args) {
        try {
            RestTemplate restTemplate = new RestTemplate();

            // Step 1: Generate Webhook
            String generateWebhookUrl =
                "https://bfhldevapigw.healthrx.co.in/hiring/generateWebhook/JAVA";

            Map<String, Object> requestBody = new HashMap<>();
            requestBody.put("name", "YOUR FULL NAME"); // <-- replace
            requestBody.put("regNo", "22BEC7233"); // <-- your regNo (odd => Q1)
            requestBody.put("email", "you@example.com");// <-- replace

            HttpHeaders headers = new HttpHeaders();
            headers.setContentType(MediaType.APPLICATION_JSON);

            HttpEntity<Map<String, Object>> entity = new HttpEntity<>(requestBody, headers);

            ResponseEntity<WebhookResponse> response =
                restTemplate.exchange(generateWebhookUrl, HttpMethod.POST, entity,
                    WebhookResponse.class);

            if (response.getStatusCode() == HttpStatus.OK && response.getBody() != null) {
```

```

String webhookUrl = response.getBody().getWebhook();
String accessToken = response.getBody().getAccessToken();

System.out.println("Webhook URL: " + webhookUrl);
System.out.println("Access Token: " + accessToken);

// Step 2: Final SQL Query (Q1/Q2 problem statement)
// Highest salary credited NOT on 1st day; include name, age, department
String finalQuery =
    "SELECT p.AMOUNT AS SALARY, " +
    "CONCAT(e.FIRST_NAME, ' ', e.LAST_NAME) AS NAME, " +
    "TIMESTAMPDIFF(YEAR, e.DOB, CURDATE()) AS AGE, " +
    "d.DEPARTMENT_NAME " +
    "FROM PAYMENTS p " +
    "JOIN EMPLOYEE e ON p.EMP_ID = e.EMP_ID " +
    "JOIN DEPARTMENT d ON e.DEPARTMENT = d.DEPARTMENT_ID " +
    "WHERE DAY(p.PAYMENT_TIME) <> 1 " +
    "ORDER BY p.AMOUNT DESC " +
    "LIMIT 1;";

Map<String, String> finalPayload = new HashMap<>();
finalPayload.put("finalQuery", finalQuery);

HttpHeaders submitHeaders = new HttpHeaders();
submitHeaders.setContentType(MediaType.APPLICATION_JSON);
submitHeaders.setBearerAuth(accessToken); // JWT token

HttpEntity<Map<String, String>> submitEntity = new HttpEntity<>(finalPayload,
submitHeaders);

// Submit to the webhook URL returned by the generator (per the instructions)
ResponseEntity<String> submitResponse =
    restTemplate.exchange(webhookUrl, HttpMethod.POST, submitEntity,
String.class);

System.out.println("Submission Response: " + submitResponse.getStatusCode());
System.out.println("Submission Body: " + submitResponse.getBody());
} else {
    System.out.println("Failed to generate webhook.");
}

} catch (Exception e) {
    e.printStackTrace();
}
}

// Helper class to parse JSON response
public static class WebhookResponse {
    @JsonProperty("webhook")
    private String webhook;

```

```
@JsonProperty("accessToken")
private String accessToken;

public String getWebhook() { return webhook; }
public void setWebhook(String webhook) { this.webhook = webhook; }
public String getAccessToken() { return accessToken; }
public void setAccessToken(String accessToken) { this.accessToken = accessToken; }
}
}
```

OUTPUT :

SALARY	NAME	AGE	DEPARTMENT_NAME
74998	Emily Brown	32	Salary