package com.example.texttoimage;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.\*;

import org.springframework.web.client.RestTemplate;

import org.springframework.http.\*;

import org.json.JSONObject;

import com.amazonaws.services.s3.AmazonS3;

import com.amazonaws.services.s3.AmazonS3ClientBuilder;

import com.amazonaws.services.s3.model.PutObjectRequest;

import com.amazonaws.auth.AWSStaticCredentialsProvider;

import com.amazonaws.auth.BasicAWSCredentials;

import java.io.File;

import java.io.FileOutputStream;

import java.nio.file.Files;

import java.util.Base64;

@SpringBootApplication

public class TextToImageApplication {

public static void main(String[] args) {

SpringApplication.run(TextToImageApplication.class, args);

}

}

@RestController

@RequestMapping("/generate-image")

class ImageGeneratorController {

private final String HUGGINGFACE\_API\_URL = "https://api-inference.huggingface.co/models/CompVis/stable-diffusion-v1-4";

private final String HUGGINGFACE\_API\_KEY = "your\_huggingface\_api\_key";

private final String AWS\_ACCESS\_KEY = "your\_aws\_access\_key";

private final String AWS\_SECRET\_KEY = "your\_aws\_secret\_key";

private final String S3\_BUCKET\_NAME = "your-s3-bucket";

private final RestTemplate restTemplate = new RestTemplate();

private final AmazonS3 s3Client;

public ImageGeneratorController() {

this.s3Client = AmazonS3ClientBuilder.standard()

.withCredentials(new AWSStaticCredentialsProvider(new BasicAWSCredentials(AWS\_ACCESS\_KEY, AWS\_SECRET\_KEY)))

.withRegion("us-east-1") // Change to your AWS region

.build();

}

@PostMapping

public ResponseEntity<?> generateImage(@RequestBody ImageRequest request) {

try {

String prompt = request.getText();

if (!"english".equalsIgnoreCase(request.getLanguage())) {

prompt = translateText(prompt, request.getLanguage());

}

String imageUrl = generateImageFromText(prompt);

String s3Url = uploadImageToS3(imageUrl);

return ResponseEntity.ok(new JSONObject().put("imageUrl", s3Url).toString());

} catch (Exception e) {

return ResponseEntity.status(HttpStatus.INTERNAL\_SERVER\_ERROR).body(e.getMessage());

}

}

private String translateText(String text, String sourceLanguage) {

String googleTranslateApiUrl = "https://translation.googleapis.com/language/translate/v2?key=your\_google\_translate\_api\_key";

HttpHeaders headers = new HttpHeaders();

headers.setContentType(MediaType.APPLICATION\_JSON);

JSONObject requestBody = new JSONObject()

.put("q", text)

.put("source", sourceLanguage)

.put("target", "en")

.put("format", "text");

HttpEntity<String> entity = new HttpEntity<>(requestBody.toString(), headers);

ResponseEntity<String> response = restTemplate.postForEntity(googleTranslateApiUrl, entity, String.class);

JSONObject jsonResponse = new JSONObject(response.getBody());

return jsonResponse.getJSONObject("data").getJSONArray("translations").getJSONObject(0).getString("translatedText");

}

private String generateImageFromText(String prompt) {

HttpHeaders headers = new HttpHeaders();

headers.set("Authorization", "Bearer " + HUGGINGFACE\_API\_KEY);

headers.setContentType(MediaType.APPLICATION\_JSON);

JSONObject requestBody = new JSONObject().put("inputs", prompt);

HttpEntity<String> entity = new HttpEntity<>(requestBody.toString(), headers);

ResponseEntity<byte[]> response = restTemplate.exchange(HUGGINGFACE\_API\_URL, HttpMethod.POST, entity, byte[].class);

byte[] imageBytes = response.getBody();

String imagePath = "generated\_image.png";

try (FileOutputStream fos = new FileOutputStream(imagePath)) {

fos.write(imageBytes);

} catch (Exception e) {

throw new RuntimeException("Failed to save image file", e);

}

return imagePath;

}

private String uploadImageToS3(String imagePath) {

File file = new File(imagePath);

String s3Key = "generated\_images/" + file.getName();

s3Client.putObject(new PutObjectRequest(S3\_BUCKET\_NAME, s3Key, file));

return "https://" + S3\_BUCKET\_NAME + ".s3.amazonaws.com/" + s3Key;

}

}

class ImageRequest {

private String text;

private String language;

public String getText() { return text; }

public void setText(String text) { this.text = text; }

public String getLanguage() { return language; }

public void setLanguage(String language) { this.language = language; }

}