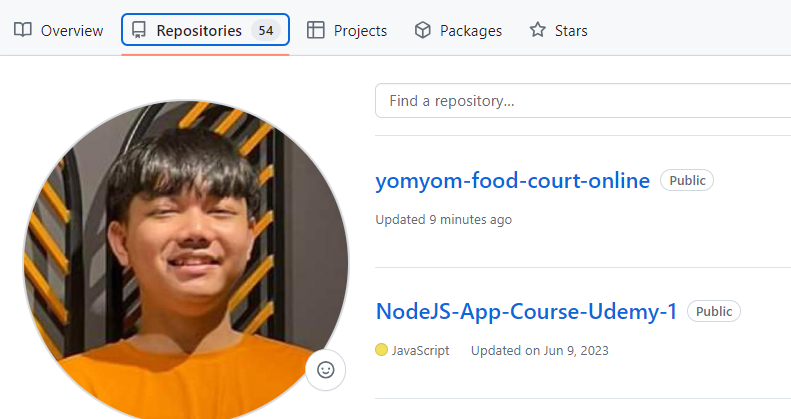
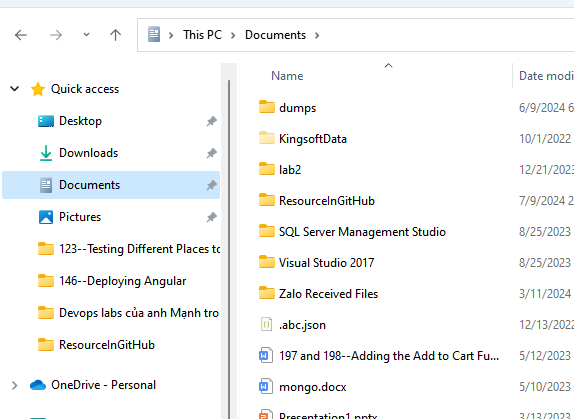
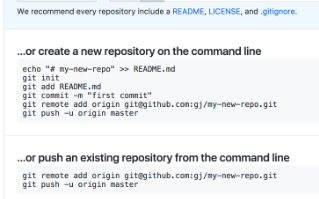
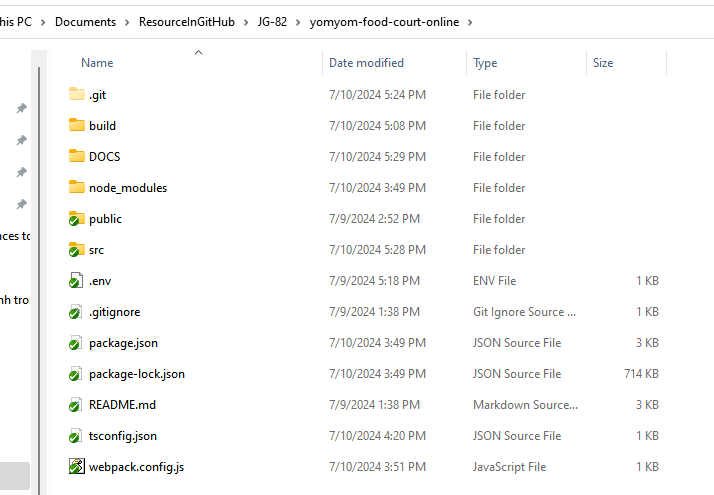
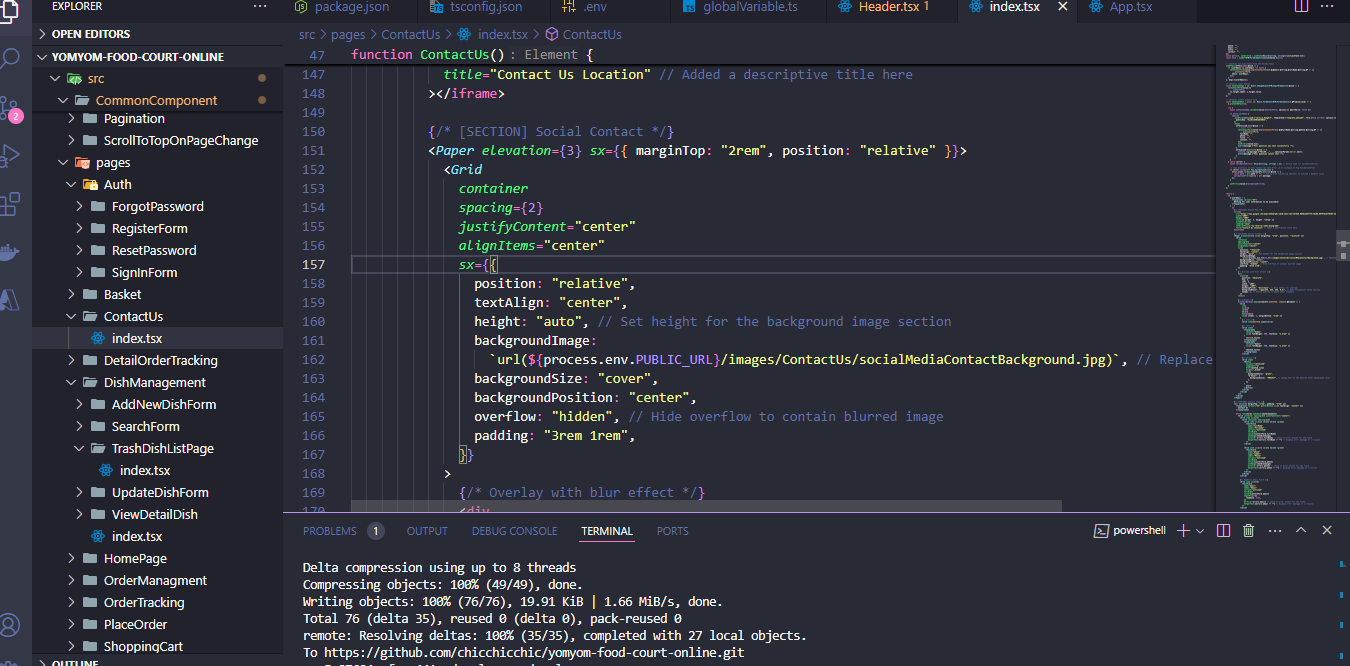
1: Create repository in Github (public):  


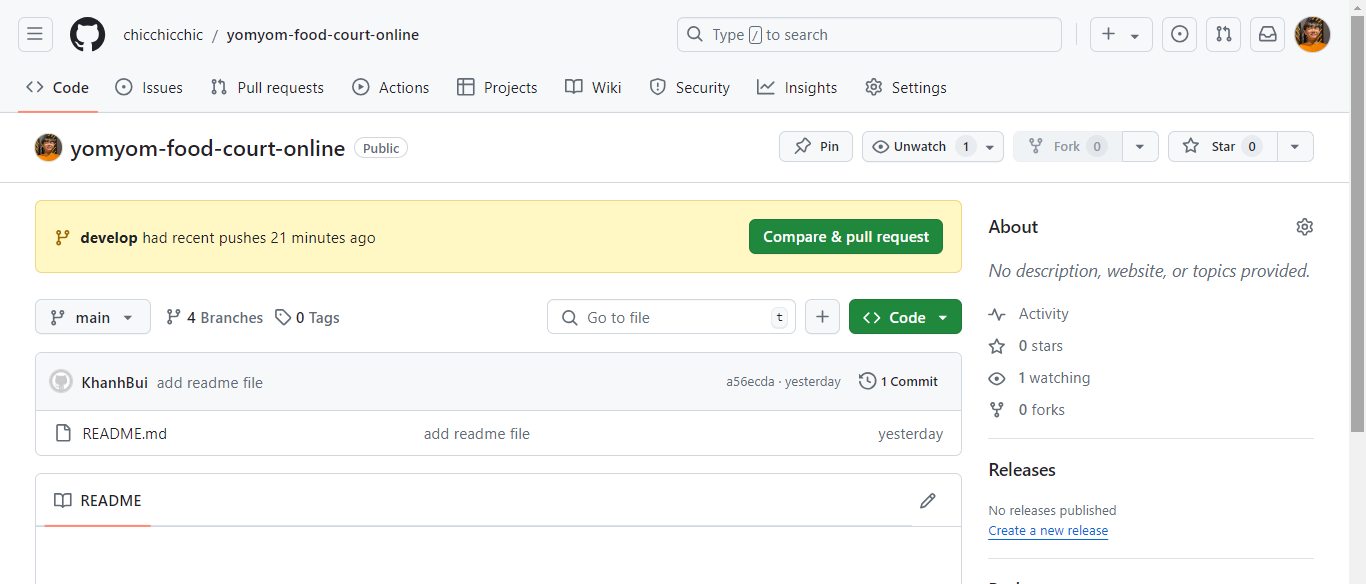
2: Create new folder With Short Path (Resolve Error: Long Path In Window):

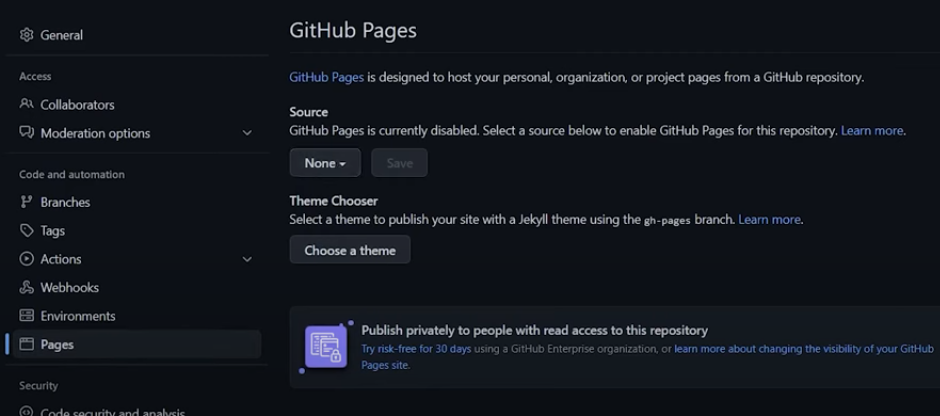
- YES: ThisPC/Document/

- NO: D:/folder1/folder2/folder3/folder4/folder5/folder6/folder7/  


3: Initial Remote Origin To The Repository:  
  
  
4: Copy All Code (not: .git file and NodeModules folder) And Paste To The Folder:  


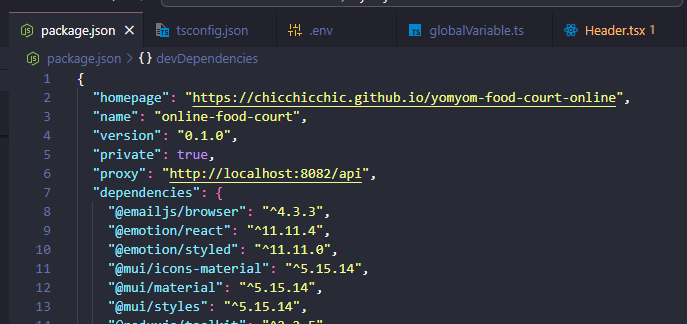
5: Open By VS Code Create Branch “Develop” From “Main”:  


6:   


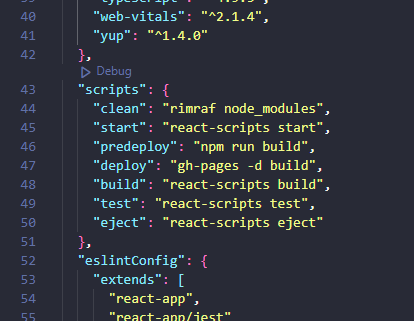


Chưa có deploy

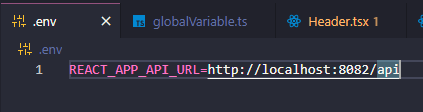
7: Install “gh-pages” in dev-dependencies:  

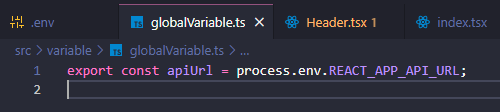

8: In package.json:  


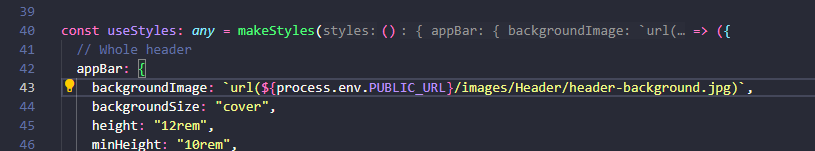
Theo cú pháp:  
https:<Username\_Github>.github.io/<Repository\_Name>

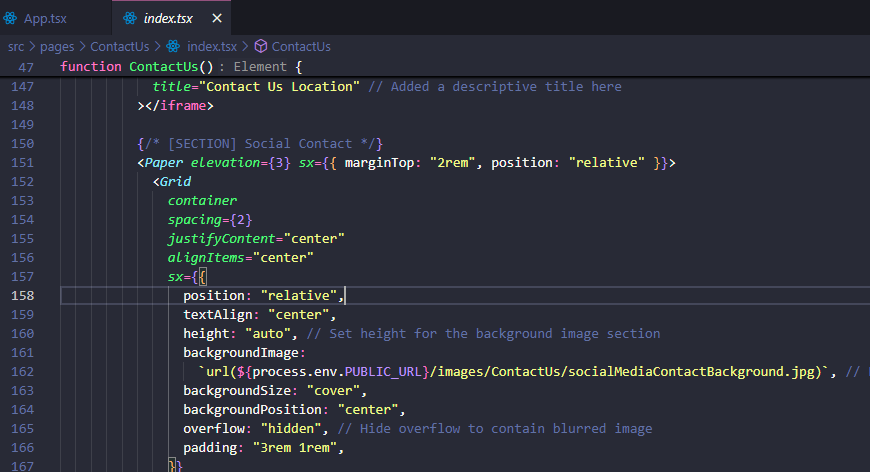


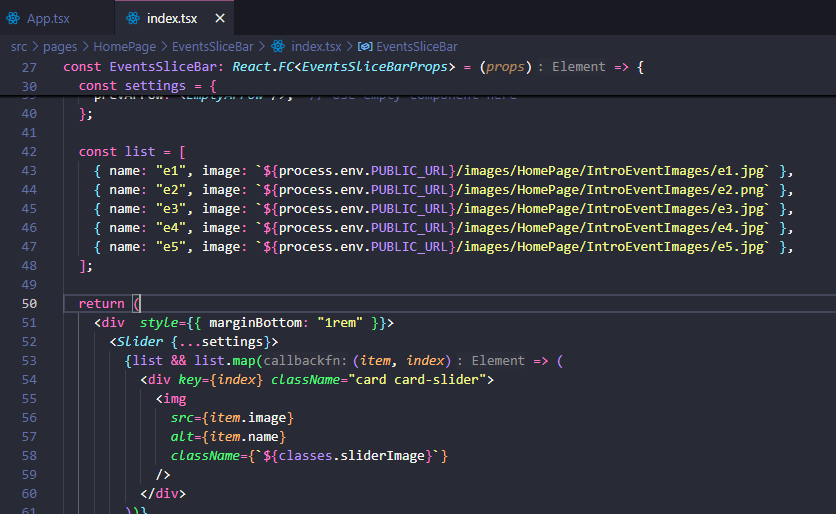
9: Create  :

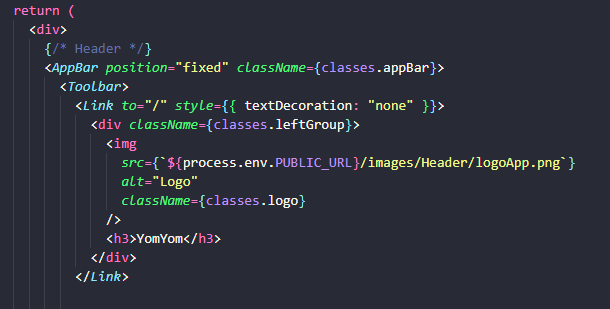
  
  
10: Create  :



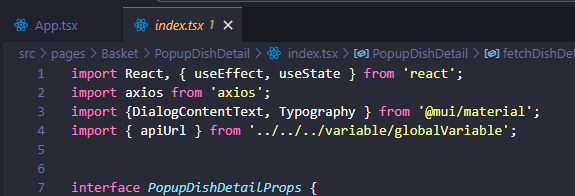
11: Replace All Image From Public Folder With New Link:  


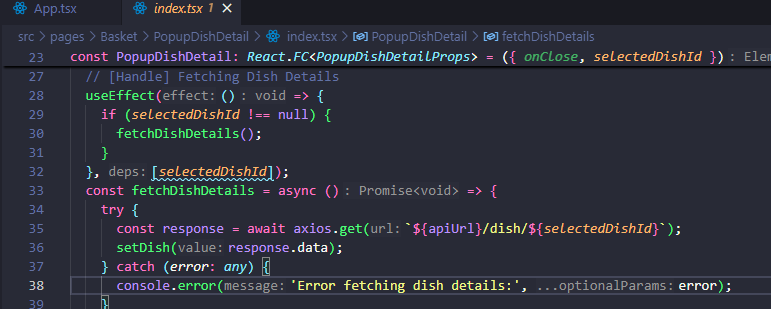






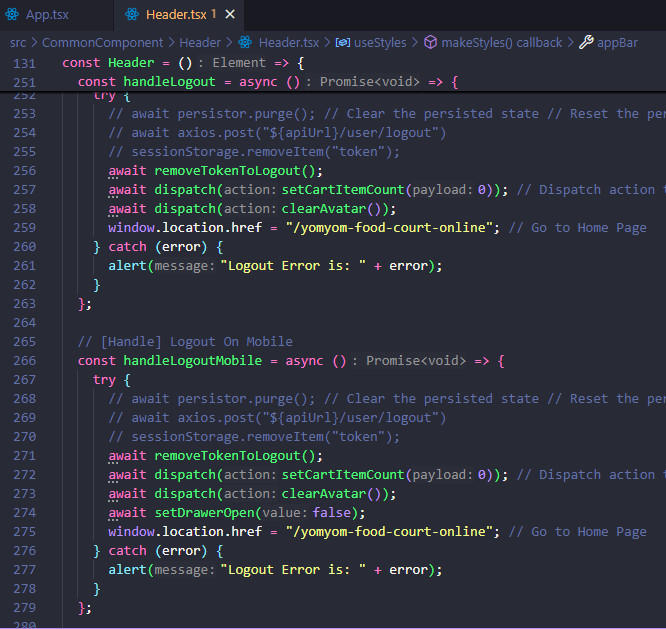
12: Replace All API Call With New API URL:



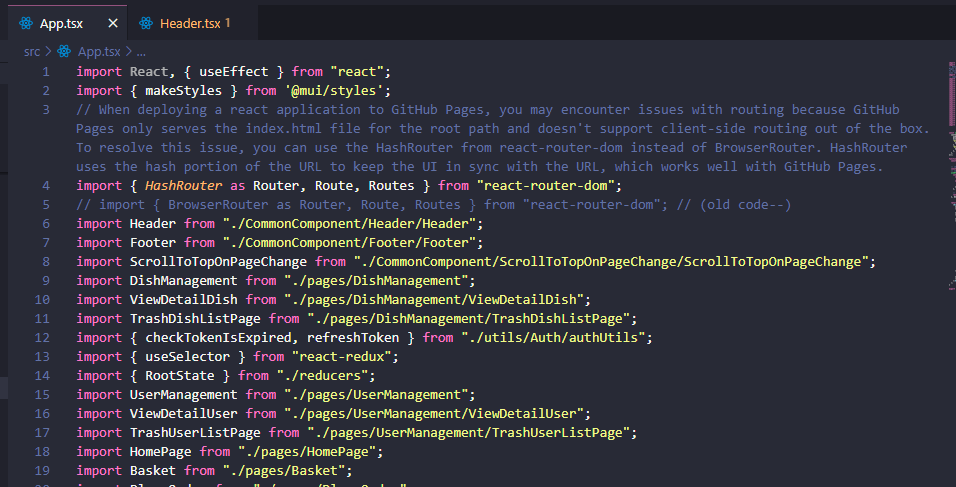


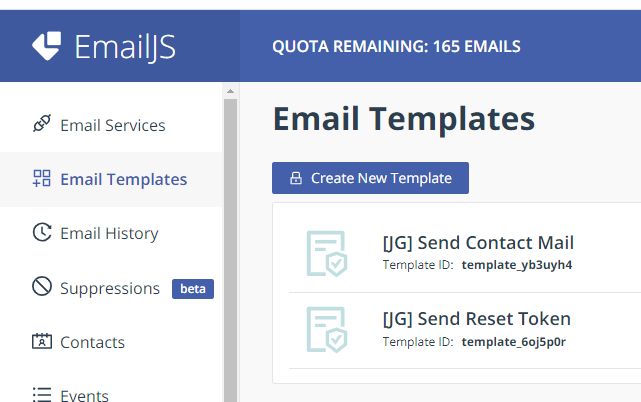


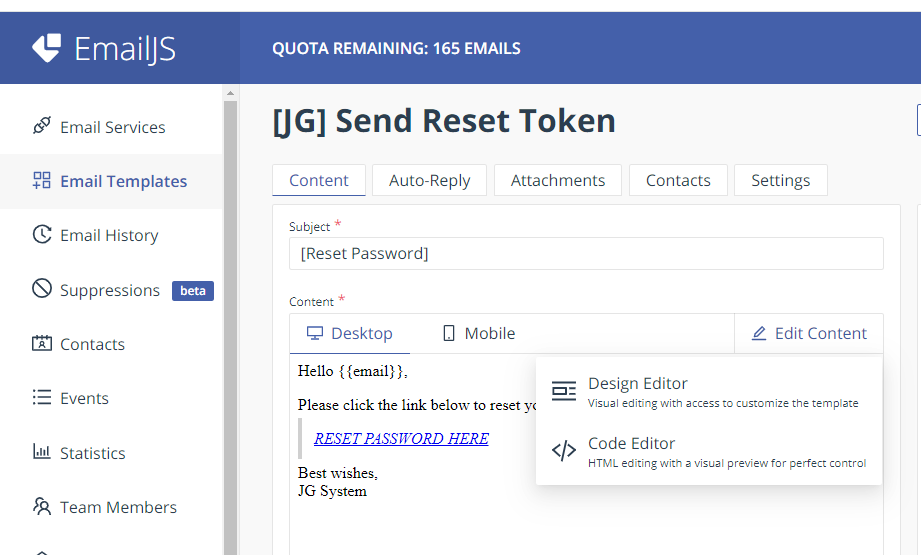
13: In Logout Handle (Destop/Tablet/Mobile) - Header Component:

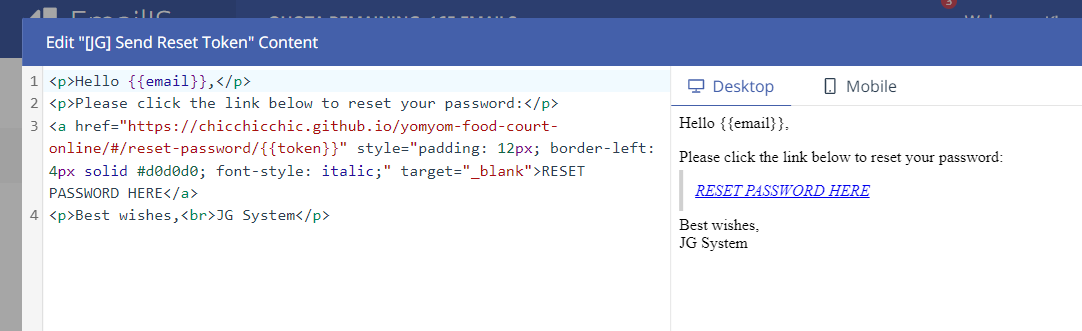


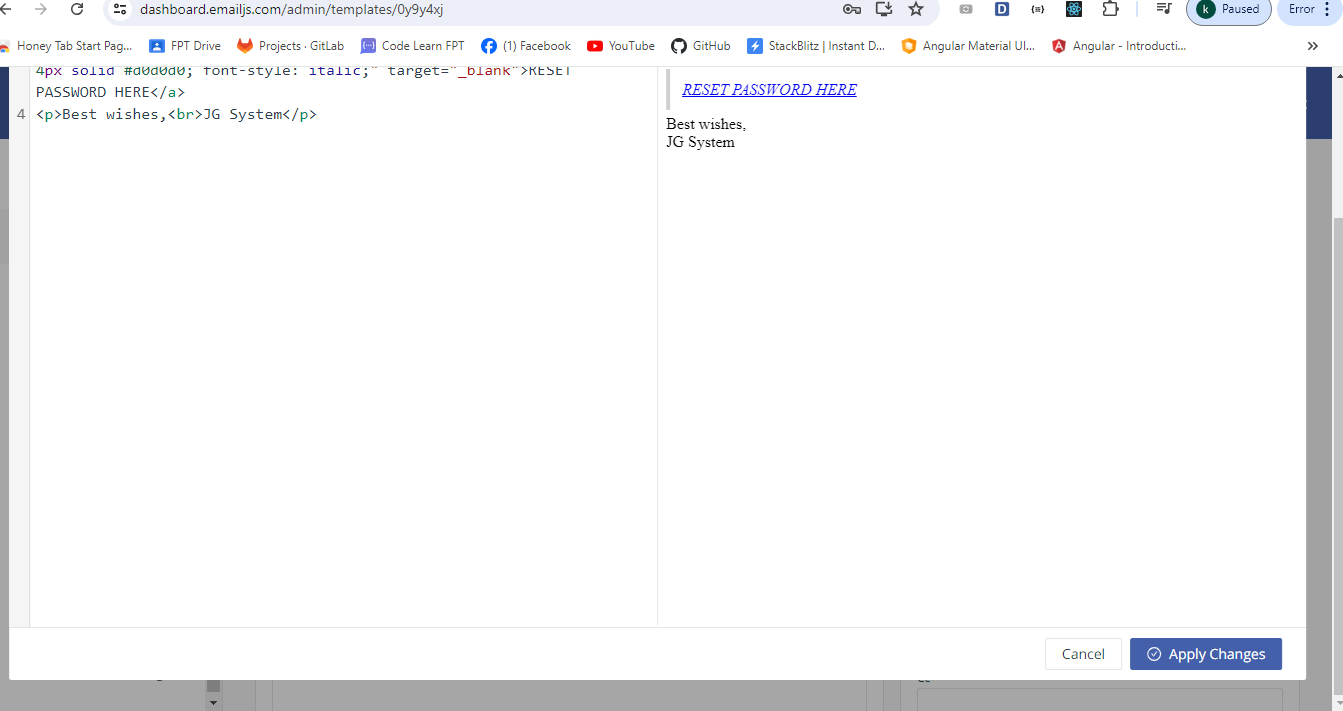
14: In App.tsx:

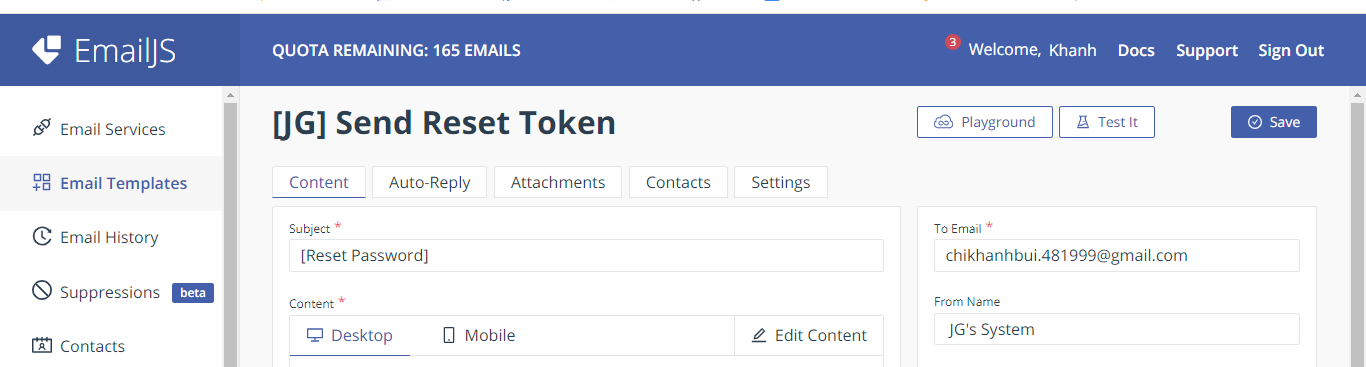


15: In EmailJS, modify Code Editer URL Of Email Template:  






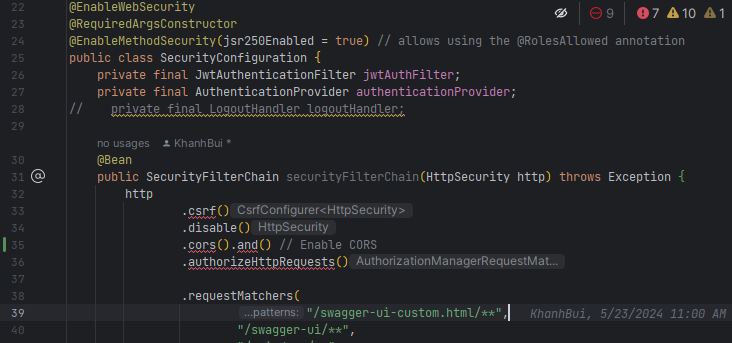




16: Create [Backend] 

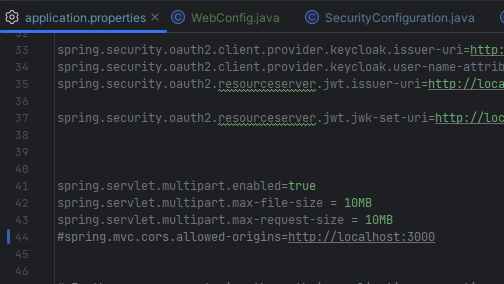
package com.jkgroup.foodCourtServerSide.configuration.WebConfig;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.cors.CorsConfiguration;  
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;  
import org.springframework.web.filter.CorsFilter;  
  
  
@Configuration  
public class WebConfig {  
 @Bean  
 public CorsFilter corsFilter() {  
 CorsConfiguration corsConfiguration = new CorsConfiguration();  
 corsConfiguration.addAllowedOrigin("https://chicchicchic.github.io"); // Specify your allowed origins  
 corsConfiguration.addAllowedHeader("\*");  
 corsConfiguration.addAllowedMethod("\*");  
 corsConfiguration.setAllowCredentials(true);  
  
 UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();  
 source.registerCorsConfiguration("/\*\*", corsConfiguration);  
  
 return new CorsFilter(source);  
 }  
}

17: In [Backend] : (Resolve CORS ERROR)



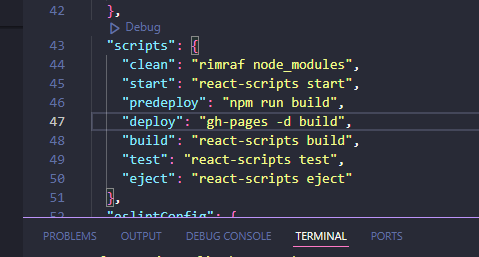


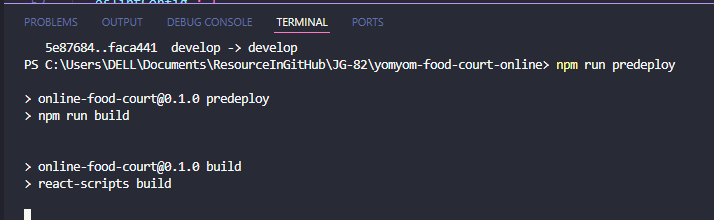
- Full Code:  
package com.jkgroup.foodCourtServerSide.configuration.SpringSecurity;  
  
import com.jkgroup.foodCourtServerSide.configuration.JWT.JwtAuthenticationFilter;  
import lombok.RequiredArgsConstructor;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.authentication.AuthenticationProvider;  
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.config.http.SessionCreationPolicy;  
import org.springframework.security.web.SecurityFilterChain;  
import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;  
import org.springframework.web.cors.CorsConfiguration;  
import org.springframework.web.cors.CorsConfigurationSource;  
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;  
  
import java.util.List;  
  
  
@Configuration  
@EnableWebSecurity  
@RequiredArgsConstructor  
@EnableMethodSecurity(jsr250Enabled = true) // allows using the @RolesAllowed annotation  
public class SecurityConfiguration {  
 private final JwtAuthenticationFilter jwtAuthFilter;  
 private final AuthenticationProvider authenticationProvider;  
// private final LogoutHandler logoutHandler;  
  
 @Bean  
 public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  
 http  
 .csrf()  
 .disable()  
 .cors().and() // Enable CORS  
 .authorizeHttpRequests()  
  
 .requestMatchers(  
 "/swagger-ui-custom.html/\*\*",  
 "/swagger-ui/\*\*",  
 "/api-docs/\*\*",  
 "/api/user/user-info/\*\*",  
 "/api/auth/\*\*",  
 "/api/dish",  
 "/api/dish/category-list",  
 "/api/dish/dishes-by-category",  
 "/api/dish/{id:\\d+}", // Only allow paths like /api/dish/1, /api/dish/2, etc.  
 "/api/reset-password/\*\*"  
 )  
 .permitAll()  
  
 .requestMatchers(  
 "/api/user/find-by-email",  
 "/api/user/find-by-email",  
 "/api/user/upload-avatar",  
 "/api/basket/\*\*",  
 "/api/order/\*\*"  
 ).hasAnyAuthority("ROLE\_MANAGER", "ROLE\_ADMIN", "ROLE\_CUSTOMER")  
  
 .requestMatchers(  
 "/api/dish/\*\*",  
 "/api/order-management/\*\*"  
 ).hasAuthority("ROLE\_MANAGER")  
 .requestMatchers(  
 "/api/demo/\*\*",  
 "/api/user/\*\*"  
 ).hasAuthority("ROLE\_ADMIN")  
  
  
 .anyRequest()  
 .authenticated()  
 .and()  
 .sessionManagement()  
 .sessionCreationPolicy(SessionCreationPolicy.*STATELESS*)  
 .and()  
 .authenticationProvider(authenticationProvider)  
 .addFilterBefore(jwtAuthFilter, UsernamePasswordAuthenticationFilter.class);  
 return http.build();  
  
 }  
  
 @Bean  
 public CorsConfigurationSource corsConfigurationSource() {  
 CorsConfiguration configuration = new CorsConfiguration();  
 configuration.setAllowedOrigins(List.*of*("https://chicchicchic.github.io")); // Specify your allowed origins  
 configuration.setAllowedMethods(List.*of*("GET", "POST", "PUT", "DELETE", "OPTIONS"));  
 configuration.setAllowedHeaders(List.*of*("Authorization", "Content-Type"));  
 configuration.setAllowCredentials(true);  
  
 UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();  
 source.registerCorsConfiguration("/\*\*", configuration);  
  
 return source;  
 }  
}

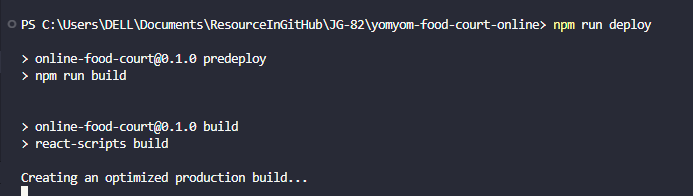
18: In application.properties:  


19: Remove All @CrossOrigin annotation In Controllers (Don’t need them anymore):  

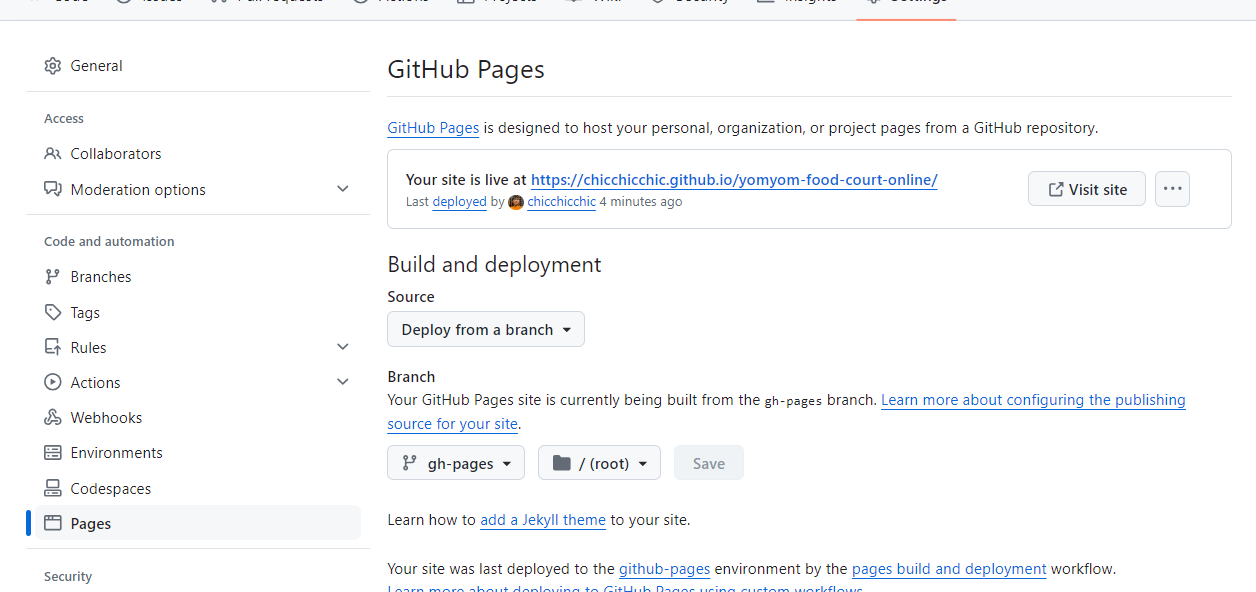

20: Build Project And Deploy Project:







21: After “npm run deploy” run successfully, wait about 2 minutes to the system deploy to server Github Pages:  
  
22. To Checking



- Click “Visit Site” Button to Check the app:

