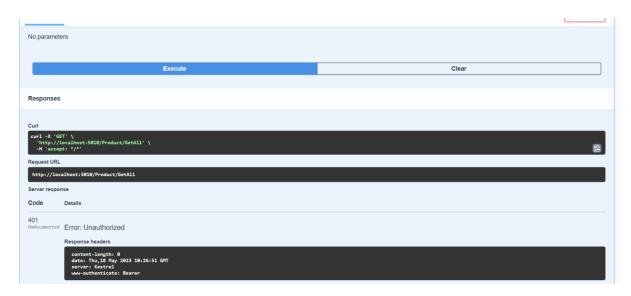
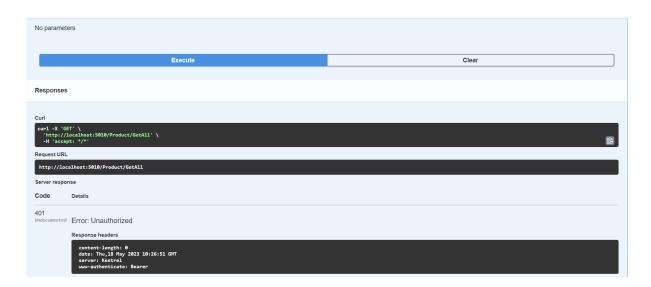
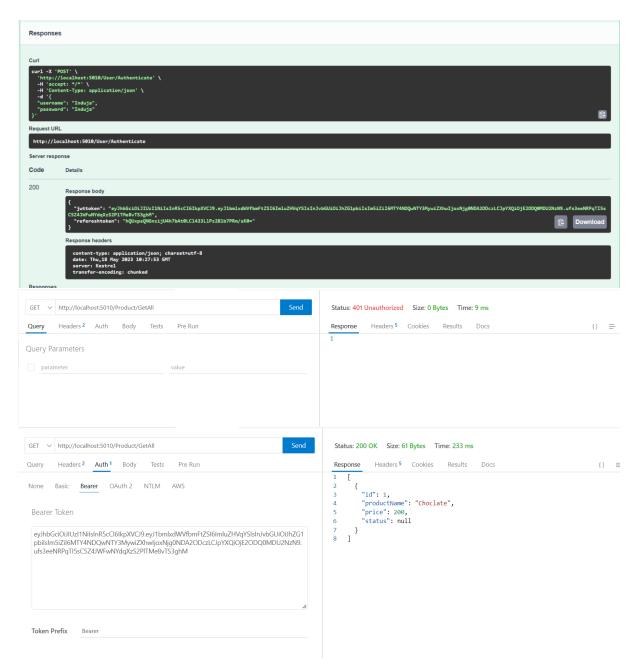
JWT AUTHENTICATION







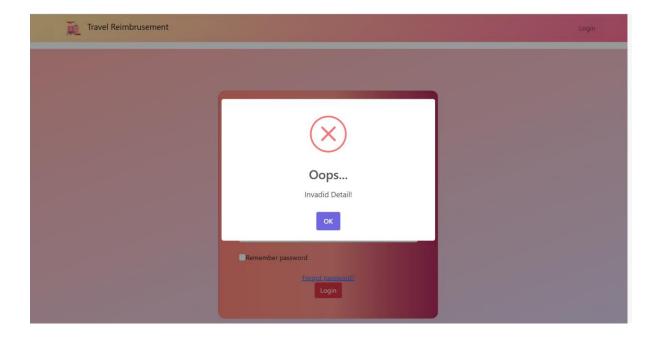
CODE

```
[HttpPost("Authenticate")]
Oreferences
public async Task<IActionResult> Authenticate([FromBody] UserCred userCred)
{
    var user = await this._DBContext.TblUsers.FirstOrDefaultAsync(item => item.Userid == userCred.username && item.Password == userCred.password if (user == null)
    | return Unauthorized();
    /// Generate Token
    var tokenhandler = new JwtSecurityTokenHandler();
    var tokenhandler = new JwtSecurityTokenHandler();
    var tokenhadler = new JwtSecurityTokenHandler();
    var tokenhadler = new SecurityTokenDescriptor
    {
        Subject = new ClaimsIdentity(
        new Claim[] { new Claim(IndimTypes.Name, user.Userid), new Claim(ClaimTypes.Role, user.Role) }
},

    Expires = DateTime.Now.AddMinutes(20),
    SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(tokenkey), SecurityAlgorithms.HmacSha256)
    };
    var token = tokenhandler.CreateToken(tokendesc);
    string finaltoken = tokenhandler.WriteToken(token);

    var response = new TokenResponse() { jwttoken = finaltoken, refereshtoken = await refereshTokenGenerator.GenerateToken(userCred.username) };
    return Ok(response);
}
```

FORMS AUTHENTICATION



ROLE BASE AUTHENTICATION

```
[HttpPost]
```