

Day 12
Dharunya S

MinisterController.cs

```
using Day13task.Context;
using Day13task.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;

namespace Day13task.Controllers
{
    [ApiController]
    [Route("[controller]")]
    public class MinisterController : Controller
    {
        MyAppDbContext appDbContext;

        public MinisterController(MyAppDbContext ctx)
        {
            appDbContext = ctx;
        }
        [Authorize(Roles = "Admin")]
        [HttpPost("InsertMinister")]
        public IActionResult AddMinister(Minister minister)
        {
            if (minister == null)
            {
                return BadRequest("Minister data is null");
            }
            appDbContext.Ministers.Add(minister);
            appDbContext.SaveChanges();
            return Ok("Minister added successfully");
        }

        [Authorize(Roles = "Officer")]

        [HttpGet("getMinister")]
        public IActionResult GetMinister()
        {
            var ministers = appDbContext.Ministers.Select(x => new { x.Id, x.State,
x.ChiefMinister, x.Language }).ToList();
            if (ministers == null )
            {
                return NotFound("No ministers found");
            }
        }
    }
}
```

```

    }
    return Ok(ministers);
}

}
}

```

UserController.cs

```

using Day13task.Context;
using Day13task.Models;
using Day13task.Security;
using Microsoft.AspNetCore.Mvc;
using System.Security.Claims;
using System.Security.Cryptography;
using System.Text;

namespace Day13task.Controllers
{
    public record LoginRequest(string Name, string Password);
    [ApiController]
    [Route("[controller]")]
    public class UserController : Controller
    {
        MyAppDbContext appDbContext;
        Microsoft.Extensions.Options.IOptions<JWTOptions> jwtOptions;

        public UserController(MyAppDbContext ctx,
Microsoft.Extensions.Options.IOptions<JWTOptions> _jwtOptions)
        {
            appDbContext = ctx;
            jwtOptions = _jwtOptions;
        }

        [HttpPost("AddUser")]
        public IActionResult AddUser(Users user)
        {
            Users user_with_hash = new Users { Name = user.Name, Email = user.Email,
Password = GetPassHash(user.Password), Role = user.Role };
            appDbContext.UsersMinister.Add(user_with_hash);
            appDbContext.SaveChanges();
            return RedirectToAction("GetAllUsers");
        }

        [HttpGet("GetAllUsers_v2")]
    }
}

```

```

public IActionResult GetAllUsers()
{
    var users = appDbContext.UsersMinister.Select(x => new { x.Name, x.Email
}).ToList();
    return Ok(users);
}

[HttpDelete("DeleteUser")]
public IActionResult DeleteUser(int Id)
{
    var user = appDbContext.UsersMinister.Where(x => x.Id == Id).FirstOrDefault();
    if (user == null)
    {
        return NotFound("User not found");
    }
    appDbContext.UsersMinister.Remove(user);
    appDbContext.SaveChanges();
    return Ok("User deleted successfully");
}

const string ADMIN = "admin";
const string OFFICER = "officer";
[HttpPost("Login")]
public IActionResult Login(LoginRequest loginReq)
{
    var name = loginReq.Name;
    var hash_pass = GetPassHash(loginReq.Password);
    var log_inuser = appDbContext.UsersMinister.Where(x => x.Name == name &&
x.Password == hash_pass).FirstOrDefault();
    if (log_inuser == null)
    {
        return Unauthorized("sorry not authorized");
    }
    else
    {
        var claims = new List<Claim>
        {
            new Claim(ClaimTypes.Name,loginReq.Name),
            //new Claim(ClaimTypes.Role,"Admin"),

        };
        if (log_inuser.Role.Trim().ToLower().Contains(ADMIN))
        {

```

```

        claims.Add(new Claim(ClaimTypes.Role, "Admin"));
    }
    if(log_inuser.Role.Trim().ToLower().Contains(OFFICER))
    {
        claims.Add(new Claim(ClaimTypes.Role, "Officer"));
    }
    var token = jwtService.CreateJwtToken(jwtOptions.Value, claims);
    return Ok(token);
}
}

```

```

private string GetPassHash(string raw_pass)
{
    using (SHA256 sha256Hash = SHA256.Create())
    {
        byte[] pass_bytes = Encoding.UTF8.GetBytes(raw_pass);
        byte[] bytes = sha256Hash.ComputeHash(pass_bytes);
        StringBuilder builder = new StringBuilder();
        for (int i = 0; i < bytes.Length; i++)
        {
            builder.Append(bytes[i].ToString("x2"));
        }
        return builder.ToString();
    }
}
}
}
}

```

[MyAppDbContext.cs](#)

```

using Microsoft.EntityFrameworkCore;
using Day13task.Models;

```

```

namespace Day13task.Context
{
    public class MyAppDbContext : DbContext
    {
        public MyAppDbContext(DbContextOptions<MyAppDbContext> options) :
        base(options) { }

        public DbSet<Minister> Ministers { get; set; }

        public DbSet<Users> UsersMinister { get; set; }
    }
}

```

```
}
```

[Minister.cs](#)

```
namespace Day13task.Models
```

```
{
```

```
    public class Minister
```

```
    {
```

```
        public int Id { get; set; }
```

```
        public string State { get; set; }
```

```
        public string ChiefMinister { get; set; }
```

```
        public string Language { get; set; }
```

```
    }
```

```
}
```

[users.cs](#)

```
using System.ComponentModel.DataAnnotations;
```

```
namespace Day13task.Models
```

```
{
```

```
    public class Users
```

```
    {
```

```
        public int Id { get; set; }
```

```
        [StringLength(25, MinimumLength = 5)]
```

```
        public string Name { get; set; }
```

```
        [StringLength(50, MinimumLength = 5)]
```

```
        public string? Email { get; set; }
```

```
        public string Password { get; set; }
```

```
        public string Role { get; set; } = "User";
```

```
    }
```

```
}
```

	Id	Name	Email	Password	Role
1	1	dharunya	dharunya@gmail.com	9e3e2d88fdd4cdefc2175899da10490fd8e0837fa1497d6b2e...	Admin
2	7	nancy	nancy@gmailc.com	1d78b370ad96e7a8091888d902360187bdea87ab6a4af003...	Officer

	Id	State	ChiefMinister	Language
1	1	Tamilnadu	Stalin	Tamil
2	2	Kerala	Pinarayi Vijayan	Malayalam

Officer token

GET http://localhost:5259/Minister/getMinister Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Auth Type
Bearer Token Token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.

Body Cookies Headers (4) Test Results (1/1) 200 OK • 90 ms • 306 B Save Response

JSON Preview Visualize

```
1 [
2   {
3     "id": 1,
4     "state": "Tamilnadu",
5     "chiefMinister": "Stalin",
6     "language": "Tamil"
7   },
8   {
9     "id": 2,
10    "state": "Kerala",
11    "chiefMinister": "Pinarayi Vijayan",
12    "language": "Malayalam"
13  }
14 ]
```

Admin token

My Collection / Post data Save Share

POST http://localhost:5259/Minister/insertMinister Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "state": "Andhra Pradesh",
3   "chiefMinister": "N. Chandrababu Naidu",
4   "language": "Telugu"
5 }
6
```

Body Cookies Headers (4) Test Results (1/1) 200 OK • 72 ms • 169 B Save Response

Raw Preview Visualize

```
1 Minister added successfully
```