



UserController
- readonly UserManager<IdentityUser> _userManager; - readonly SignInManager<IdentityUser> _signInManager; - readonly ApplicationDbContext _applicationDbContext;
+ UserController(UserManager<IdentityUser> userManager, SignInManager<IdentityUser> signInManager, ApplicationDbContext applicationDbContext) + IActionResult Index() + async Task<IActionResult> Create() [HttpPost] + async Task<IActionResult> Create(AppUser user) + async Task<IActionResult> Details(string id) + async Task<IActionResult> Delete(string id) [HttpPost] + async Task<IActionResult> Delete(AppUser appUser)
AdvertisementController
- ApplicationDbContext _db; - readonly UserManager<IdentityUser> _userManager; - readonly SignInManager<IdentityUser> _signInManager;
+ AdvertisementController(ApplicationDbContext db, SignInManager<IdentityUser> signInManager, UserManager<IdentityUser> userManager) + IActionResult Index() [HttpPost] public IActionResult Index(decimal? price, string? carName,int? manufacturingYear,int? buldingNumber,string? street,string? area, string? city,string? governorate) + IActionResult Create() [HttpPost] + async Task<IActionResult> Create(AdvertisementtViewModel advertisement) + async Task<IActionResult> DetailsAsync(int? id) + async Task<IActionResult> ReserveAsync(string? reserverId, int? id) + async Task<IActionResult> GetReservations() + async Task<IActionResult> Delete(string? userId, int? id)
Login
- readonly SignInManager<IdentityUser> _signInManager; - readonly ApplicationDbContext _context; - readonly UserManager<IdentityUser> _userManager; [BindProperty] + InputModel Input { get; set; } + IList<AuthenticationScheme> ExternalLogins { get; set; } + string returnUrl { get; set; } [TempData] + public string ErrorMessage { get; set; }
+ LoginModel(SignInManager<IdentityUser> signInManager, ApplicationDbContext context, UserManager<IdentityUser> userManager) + async Task OnGetAsync(string returnUrl = null) + async Task<IActionResult> OnPostAsync(string returnUrl = null)
Register
- readonly SignInManager<IdentityUser> _signInManager - readonly UserManager<IdentityUser> _userManager - readonly IEmailSender _emailSender + [BindProperty] + InputModel Input { get; set; } + string returnUrl { get; set; } + IList<AuthenticationScheme> ExternalLogins { get; set; }
+ RegisterModel(UserManager<IdentityUser> userManager,SignInManager<IdentityUser> signInManager, IEmailSender emailSender) + public async Task OnGetAsync(string returnUrl = null) + public async Task<IActionResult> OnPostAsync(string returnUrl = null) - IdentityUser CreateUser()
ApplicationDbContext
+ DbSet<Advertisement> Advertisements { get; set; } + DbSet<AppUser> AppUsers { get; set; } + DbSet<IdentityUserRole<string>> UserRoles { get; set; }  + public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options) + protected override void OnModelCreating(ModelBuilder builder)

AppUser
<div>[Required(ErrorMessage = "Invalid Input")] [StringLength(8, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[a-zA-Z]+\$", ErrorMessage = "Invalid Input")] + string FirstName { get; set; }  [Required(ErrorMessage = "Invalid Input")] [StringLength(8, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[a-zA-Z]+\$", ErrorMessage = "Invalid Input")] + string LastName { get; set; } [Required(ErrorMessage = "Invalid Input")] [StringLength(30, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[a-zA-Z0-9\s.]+\$", ErrorMessage = "Invalid Input")] + string Address { get; set; }  [Required(ErrorMessage = "Invalid Input")] [Range(18, 120, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[0-9]+\$", ErrorMessage = "Invalid Input")] + int Age { get; set; }]</div>

InputModel [For Login]
<div>[Required] [EmailAddress] public string Email { get; set; }  [Required(ErrorMessage = "Invalid Input")] [StringLength(20, MinimumLength = 8, ErrorMessage = "Invalid Input")] [RegularExpression(@"^(?=[a-zA-Z])(?=[\d])(?=[^\da-zA-Z]).{8,20}\$", ErrorMessage = "Invalid Input")] [DataType(DataType.Password)] public string Password { get; set; }  [Display(Name = "Remember me?")] public bool RememberMe { get; set; }]</div>

InputModel [For Register]
<div>[Required(ErrorMessage = "Invalid Input")] [EmailAddress(ErrorMessage = "Invalid Input")] [Display(Name = "Email")] + string Email { get; set; } [Required(ErrorMessage = "Invalid Input")] [StringLength(20, MinimumLength = 8, ErrorMessage = "Invalid Input")] [RegularExpression(@"^(?=[a-zA-Z])(?=[\d])(?=[^\da-zA-Z]).{8,20}\$", ErrorMessage = "Invalid Input")] [DataType(DataType.Password, ErrorMessage = "Invalid Input")] + string Password { get; set; }  [DataType(DataType.Password)] [Display(Name = "Confirm password")] [Compare("Password", ErrorMessage = "Invalid Input")] + string ConfirmPassword { get; set; }  [Required(ErrorMessage = "Invalid Input")] [StringLength(8, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[a-zA-Z]+\$", ErrorMessage = "Invalid Input")] + string FirstName { get; set; }  [Required(ErrorMessage = "Invalid Input")] [StringLength(8, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[a-zA-Z]+\$", ErrorMessage = "Invalid Input")] + string LastName { get; set; }  [Required(ErrorMessage = "Invalid Input")] [StringLength(30, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[a-zA-Z0-9\s.]+\$", ErrorMessage = "Invalid Input")] + string Address { get; set; }  [Required(ErrorMessage = "Invalid Input")] [Range(18, 120, ErrorMessage = "Invalid Input")] [RegularExpression(@"^[0-9]+\$", ErrorMessage = "Invalid Input")] + int Age { get; set; }  [Required(ErrorMessage = "Invalid Input")] [StringLength(12, ErrorMessage = "Invalid Input")] [RegularExpression(@"^20[0-9]{10}\$", ErrorMessage = "Invalid Input")] + string PhoneNumber { get; set; }]</div>

## Advertisermt

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

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

```
[Required(ErrorMessage = "Invalid Input")]  
[StringLength(30, MinimumLength = 3, ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^[a-zA-Z][a-zA-Z0-9\s]*$", ErrorMessage = "Invalid Input")]  
[DisplayName("Car Name")]  
public String CarName { get; set; }
```

```
[DisplayName("Manufacturing Year")]  
[Required(ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^(19[0-9]{2}|20[0-9]{2})$", ErrorMessage = "Invalid Input")]  
[Range(1900, 2023, ErrorMessage = "Invalid Input")]  
public int? ManufacturingYear { get; set; }
```

```
[DisplayName("Price")]  
[Required(ErrorMessage = "Invalid Input")]  
[Range(100000, long.MaxValue, ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^[1-9]\d*$", ErrorMessage = "Invalid Input")]  
public int Price { get; set; }
```

```
[DisplayName("Phone Number")]  
[Required(ErrorMessage = "Invalid Input")]  
[StringLength(12, MinimumLength = 12, ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^20[1-9]\d{8}$", ErrorMessage = "Invalid Input")]  
public String PhoneNumber { get; set; }
```

```
[DisplayName("Licence Expiration Date")]
```

```
public DateTime LicenceExpirationDate { get; set; }  
public string Image { get; set; }
```

```
[DisplayName("Bulding Number")]  
[Required(ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^[1-9]\d{0,3}$", ErrorMessage = "Invalid Input")]  
public int? BuldingNumber { get; set; }
```

```
[Required(ErrorMessage = "Invalid Input")]  
[StringLength(20, MinimumLength = 3, ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^[a-zA-Z][a-zA-Z ]*[a-zA-Z]$", ErrorMessage = "Invalid Input")]  
public String Street { get; set; }
```

```
[Required(ErrorMessage = "Invalid Input")]  
[StringLength(20, MinimumLength = 3, ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^[a-zA-Z][a-zA-Z ]*[a-zA-Z]$", ErrorMessage = "Invalid Input")]  
public String Area { get; set; }
```

```
[Required(ErrorMessage = "Invalid Input")]  
[StringLength(20, MinimumLength = 3, ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^[a-zA-Z][a-zA-Z ]*[a-zA-Z]$", ErrorMessage = "Invalid Input")]  
public String City { get; set; }
```

```
[Required(ErrorMessage = "Invalid Input")]  
[StringLength(20, MinimumLength = 3, ErrorMessage = "Invalid Input")]  
[RegularExpression(@"^[a-zA-Z][a-zA-Z ]*[a-zA-Z]$", ErrorMessage = "Invalid Input")]  
public String Governorate { get; set; }
```

-