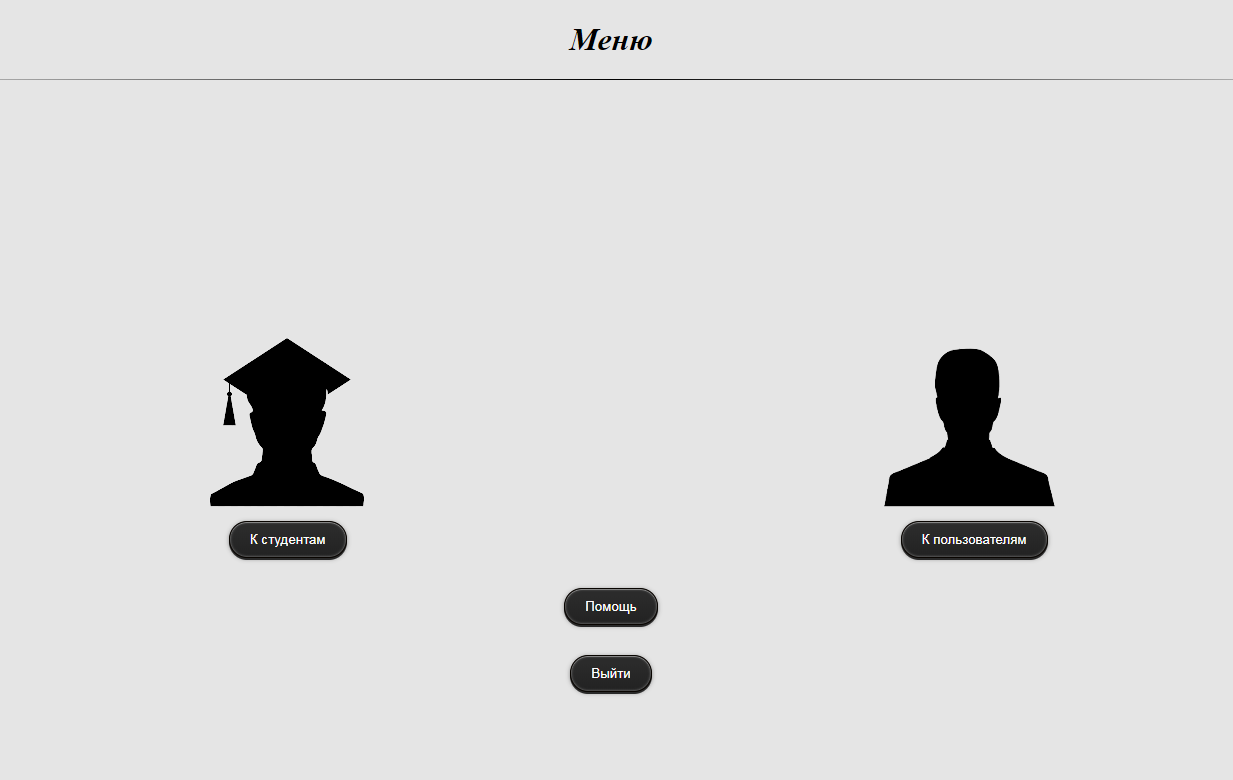
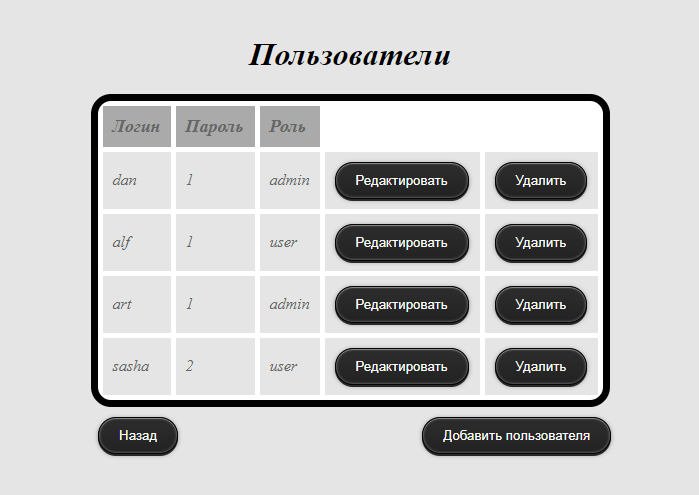
**Лабораторная работа №6.**

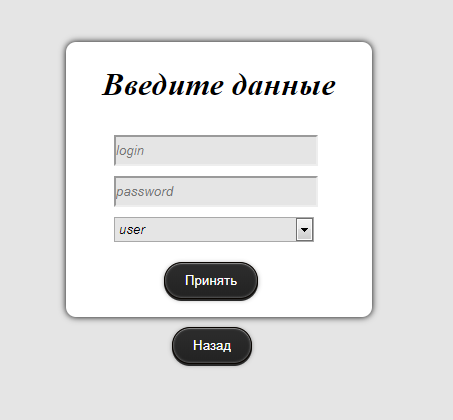
**Технология Model-View-Controller**

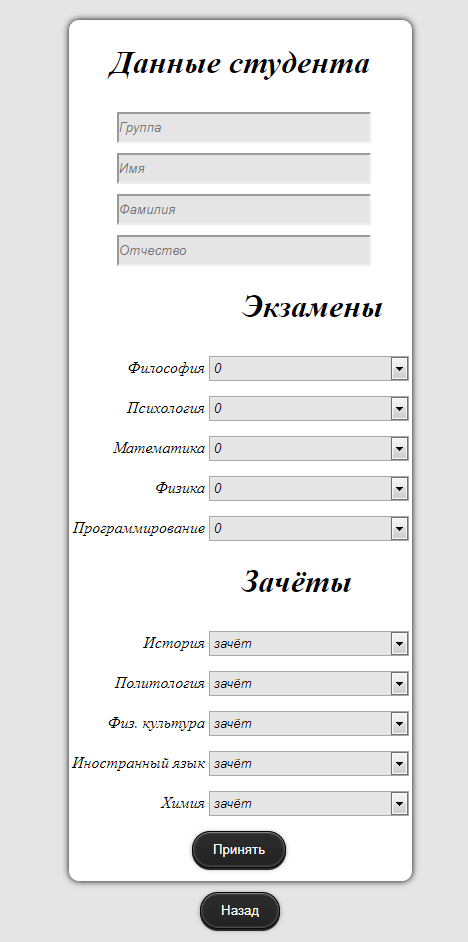
**Самусев Д. А. АСОИ-181**

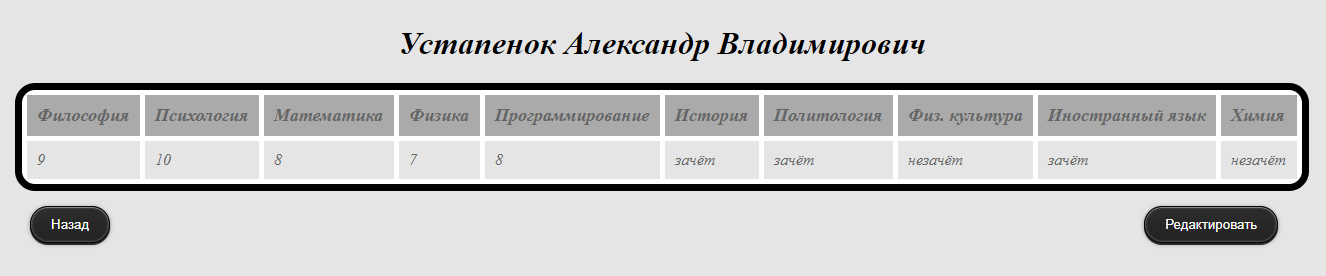
**Цель работы.** Познакомиться с технологией ASP.Net











UserController.cs

**namespace** AccountingSystem.Controllers

{ **public class** UsersController : Controller  
 {  
 **private readonly** IUserRepository \_userRepository;  
 **private readonly** Validator \_validator;  
 **private readonly** AbstractValidator<User> \_userValidator;  
 **private const string** *Users\_* = "users";  
 **private const string** *UserExists* = "Пользователь с таким логином уже существует!";  
  
 **public** UsersController(IUserRepository userRepository, Validator validator,  
 AbstractValidator<User> userValidator)  
 {  
 \_userRepository = userRepository;  
 \_validator = validator;  
 \_userValidator = userValidator;  
 }  
  
 **public** IActionResult Users()  
 {  
 IActionResult result = CheckUserAccess();  
   
 **if** (result != **null**)  
 {  
 **return** result;  
 }  
   
 List<User> users = GetUsers();  
   
 **return** View(users);  
 }  
  
 **public** IActionResult DeleteUser(**long** userId)  
 {   
 IActionResult result = CheckUserAccess();  
   
 **if** (result != **null**)  
 {  
 **return** result;  
 }  
  
 **if** (userId < 0)  
 {  
 **return** View("~/Views/Error400.cshtml");  
 }  
   
 List<User> users = GetUsers();  
  
 \_userRepository.DeleteOneById(userId);  
 User user = users.FirstOrDefault(u => u.Id == userId);  
 users.Remove(user);  
 HttpContext.Session.Set(*Users\_*, users);  
  
 **return** RedirectToAction("UsersResult", "Users", **new** {message = "Пользователь успешно удалён"});  
 }  
  
 **public** IActionResult ModifyUser(User user)  
 {   
 IActionResult actionResult = CheckUserAccess();  
   
 **if** (actionResult != **null**)  
 {  
 **return** actionResult;  
 }  
   
 ValidationResult result = \_userValidator.Validate(user);  
  
 **if** (!result.IsValid)  
 {  
 **return** View("~/Views/Error400.cshtml");  
 }  
   
 List<User> users = GetUsers();  
  
 **if** (\_validator.UserIsUnique(user ,users))  
 {  
 User oldUser = users.FirstOrDefault(u => u.Id == user.Id);  
  
 \_userRepository.Modify(user);  
 users.Remove(oldUser);  
 users.Add(user);  
 HttpContext.Session.Set(*Users\_*, users);   
 HttpContext.Session.Set("userModifyError", "");  
 }  
 **else** {  
 **string** modifyError = *UserExists*;  
 HttpContext.Session.Set("userModifyError", modifyError);  
 }  
   
 **return** RedirectToAction("UsersResult", "Users", **new** {message = "Пользователь успешно обновлён"});  
 }  
  
 [HttpGet]  
 **public** IActionResult AddUser()  
 {   
 IActionResult result = CheckUserAccess();  
   
 **if** (result != **null**)  
 {  
 **return** result;  
 }  
   
 HttpContext.Session.Set("userAddError", "");   
 **return** View();  
 }  
   
 [HttpPost]  
 **public** IActionResult AddUser(User user)  
 {  
 IActionResult actionResult = CheckUserAccess();  
   
 **if** (actionResult != **null**)  
 {  
 **return** actionResult;  
 }  
  
 ValidationResult result = \_userValidator.Validate(user);  
  
 **if** (!result.IsValid)  
 {  
 **return** View("~/Views/Error400.cshtml");  
 }  
   
 List<User> users = GetUsers();  
  
 **if** (\_validator.UserIsUnique(user ,users))  
 {  
 \_userRepository.Add(user);  
 users.Add(user);  
 HttpContext.Session.Set(*Users\_*, users);  
 HttpContext.Session.Set("userAddError", "");   
 }  
 **else** {  
 **string** userAddError = *UserExists*;  
 HttpContext.Session.Set("userAddError", userAddError);  
 **return** View();  
 }  
  
 **return** RedirectToAction("UsersResult", "Users", **new** {message = "Пользователь успешно добавлен"});  
 }  
  
 **public** IActionResult UsersResult(**string** message)  
 {  
 **return** View(model: message);  
 }  
   
 **private** List<User> GetUsers()  
 {  
 List<User> users = HttpContext.Session.Get<List<User>>(*Users\_*) ?? \_userRepository.GetAll();  
  
 **return** users;  
 }  
  
 **public** IActionResult CheckUserAccess()  
 {  
 User currentUser = HttpContext.Session.Get<User>("user");  
  
 **if** (currentUser.Role != "admin")  
 {  
 **return** View("AccessError");  
 }  
  
 **return null**;  
 }  
 }

}

|  |
| --- |
| namespace AccountingSystem.Models |
|  | { |
|  | public class User |
|  | { |
|  | public long Id { get; set; } |
|  | public string Login { get; set; } |
|  | public int Password { get; set; } |
|  | public string Role { get; set; } |
|  | } |
|  | } |

|  |  |
| --- | --- |
| @using AccountingSystem.Models | |
|  | | @using AccountingSystem.Extensions; |
|  | | @model List<User> |
|  | | @{ |
|  | | string error = Context.Session.Get<string>("userModifyError"); |
|  | | } |
|  | | <!DOCTYPE html> |
|  | |  |
|  | | <html> |
|  | | <head> |
|  | | <title>Users</title> |
|  | | <link href="~/css/common.css" type="text/css" rel="stylesheet"/> |
|  | | <link href="~/css/Users/users.css" type="text/css" rel="stylesheet"/> |
|  | | <script src="~/js/users.js" type="text/javascript"></script> |
|  | | </head> |
|  | | <body> |
|  | |  |
|  | | @if (error != null) |
|  | | { |
|  | | <div> |
|  | | <h1>@error</h1> |
|  | | </div> |
|  | | } |
|  | |  |
|  | | <div id="modifyWindow"> |
|  | | <img class="closeImage" onclick="modifyUser('none')" src="~/images/close.png"> |
|  | | <h1>Введите новые данные</h1> |
|  | |  |
|  | | <form action="~/Users/ModifyUser"> |
|  | | <input type="hidden" id="modifyId" name="Id"/> |
|  | | <input required placeholder="login" name="Login"/><br/> |
|  | | <input required class="passwordInput" placeholder="password" name="Password"/><br/> |
|  | | <select size="1" class="roleInput" name="Role"> |
|  | | <option value="user">user</option> |
|  | | <option value="admin">admin</option> |
|  | | </select><br/> |
|  | | <button class="submitButton" type="submit">Принять</button> |
|  | | </form> |
|  | | </div> |
|  | |  |
|  | | <div id="deleteWindow"> |
|  | | <img class="closeImage" onclick="deleteUser('none')" src="~/images/close.png"> |
|  | | <h1>Вы уверены?</h1> |
|  | |  |
|  | | <form action="~/Users/DeleteUser"> |
|  | | <input type="hidden" id="deleteId" name="userId"/> |
|  | | <button class="submitButton" type="submit">Принять</button> |
|  | | </form> |
|  | | </div> |
|  | |  |
|  | | <h1 class="mainH1">Пользователи</h1> |
|  | |  |
|  | | <div> |
|  | | <table> |
|  | | <tr> |
|  | | <th>Логин</th> |
|  | | <th>Пароль</th> |
|  | | <th>Роль</th> |
|  | | </tr> |
|  | | @foreach (User user in Model) |
|  | | { |
|  | | <tr> |
|  | | <td>@user.Login</td> |
|  | | <td>@user.Password</td> |
|  | | <td>@user.Role</td> |
|  | | <td> |
|  | | <button onclick="modifyUser('block', @user.Id)">Редактировать</button> |
|  | | </td> |
|  | | <td> |
|  | | <button onclick="deleteUser('block', @user.Id)">Удалить</button> |
|  | | </td> |
|  | | </tr> |
|  | | } |
|  | | </table> |
|  | | </div> |
|  | |  |
|  | | <div> |
|  | | <form action="~/Menu/Menu"> |
|  | | <button class="backButton" type="submit">Назад</button> |
|  | | </form> |
|  | |  |
|  | | <form method="get" action="~/Users/AddUser"> |
|  | | <button class="addUserButton" type="submit">Добавить пользователя</button> |
|  | | </form> |
|  | | </div> |
|  | |  |
|  | | </body> |
|  | | </html> |
| @using AccountingSystem.Extensions |
|  | @{ |
|  | string error = Context.Session.Get<string>("userAddError"); |
|  | } |
|  |  |
|  | <!DOCTYPE html> |
|  |  |
|  | <html> |
|  | <head> |
|  | <title>title</title> |
|  | <link href="~/css/common.css" type="text/css" rel="stylesheet"/> |
|  | <link href="~/css/Users/addUser.css" type="text/css" rel="stylesheet"/> |
|  | </head> |
|  | <body> |
|  |  |
|  | @{ |
|  | if (error != null) |
|  | { |
|  | <h1>@error</h1> |
|  | } |
|  | } |
|  |  |
|  | <form class="inputForm" method="post" action="~/Users/AddUser"> |
|  | <h1>Введите данные</h1> |
|  | <input required placeholder="login" name="Login"/><br/> |
|  | <input required class="passwordInput" placeholder="password" name="Password"/><br/> |
|  | <select size="1" class="roleInput" name="Role"> |
|  | <option value="user">user</option> |
|  | <option value="admin">admin</option> |
|  | </select><br/> |
|  | <button class="submitButton" type="submit">Принять</button> |
|  | </form> |
|  |  |
|  | <form method="get" action="~/Users/Users"> |
|  | <button class="backButton" type="submit">Назад</button> |
|  | </form> |
|  |  |
|  | </body> |
|  | </html> |