```
namespace ELearningProject.Controllers
{
    public class UserController : Controller
    {

        private readonly ApplicationDbContext db = new ApplicationDbContext();

        [HttpGet]
        public ActionResult Register()
        {
            return View();
        }


        [HttpPost]
        public ActionResult Register(RegisterVM obj)
        {
            Session["UserId"] = null;
            bool UserExistis = db.User.Any(x => x.Username == obj.Username);
            if (UserExistis)
            {
                ViewBag.UserNameMessage = "Perdoruesi egziston, provoni tjeter";
                return View();
            }
            bool EmailExistis = db.User.Any(y => y.Email == obj.Email);
            if (EmailExistis)
            {
                ViewBag.EmailMessage = "Email egziston, provoni nje email tjeter";
                return View();
    }
    User u = new User
    {
        Emri = obj.Emri,
        Mbiemri = obj.Mbiemri,
        Username = obj.Username,
        Password = obj.Password,
        Email = obj.Email,
        ImageUrl = "",
        CreatedOn = DateTime.Now,
        RoleId = 3
    };

    db.User.Add(u);
    db.SaveChanges();
    return RedirectToAction("Login", "User");
}


[HttpGet]
public ActionResult Login()
{
    Session["UserId"] = null;
    return View();
}


[HttpPost]
public ActionResult Login(LoginVM obj)
```

```
{

    bool exist = db.User.Any(u => u.Username == obj.Username && u.Password == obj.Password);
    if (!exist)
    {
        ViewBag.Error = "Perdoruesi nuk egziston";
        return View();
    }
    int userRole = db.User.Single(x => x.Username == obj.Username).RoleId;
    if (userRole == 1)
    {
        Session["Username"] = db.User.Single(x => x.Username == obj.Username).Username;
        Session["UserId"] = db.User.Single(x => x.Username == obj.Username).Id;
        Session["ImageUrl"] = db.User.Single(x => x.Username == obj.Username).ImageUrl;
        return RedirectToAction("Index", "UserHome");
    }
    else if (userRole == 2)
    {
        Session["Username"] = db.User.Single(x => x.Username == obj.Username).Username;
        Session["UserId"] = db.User.Single(x => x.Username == obj.Username).Id;
        Session["ImageUrl"] = db.User.Single(x => x.Username == obj.Username).ImageUrl;
        return RedirectToAction("Index", "UserInstruktor");
    }
    else
    {
        Session["Username"] = db.User.Single(x => x.Username == obj.Username).Username;
        Session["UserId"] = db.User.Single(x => x.Username == obj.Username).Id;
        Session["ImageUrl"] = db.User.Single(x => x.Username == obj.Username).ImageUrl;
        return RedirectToAction("Index", "UserAdmin");
    }
}



[HttpGet]
public ActionResult Logout()
{
    Session["UserId"] = 0;
    return RedirectToAction("Login", "User");
}



public ActionResult ChangePassword()
{
    int userId = Convert.ToInt32(Session["UserId"]);
    //Useri nuk eshte loguar
    if (userId == 0)
    {
        return RedirectToAction("Login", "User");
    }
    return View();
}



[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult ChangePassword(ChangePasswordVM obj)
{
    bool exist = db.User.Any(x => x.Password == obj.OldPassword);
    if (!exist)
    {
        ModelState.AddModelError("OldPassword", "Passwordi nuk eshte i sakte");
    }
    else
    {
        int userId = Convert.ToInt32(Session["UserId"]);
        User user = db.User.Find(userId);
        user.Password = obj.ConfirmPassword;
        db.Entry(user).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index", "Home");
    }
    return View();
}


public ActionResult UserProfile()
{
    int userId = Convert.ToInt32(Session["UserId"]);
    if (userId == 0)
    {
        return RedirectToAction("Login", "User");
    }
    return View(db.User.Find(userId));
}
```

```
[HttpPost]
public ActionResult UpdatePicture(PictureVM obj)
{
   int userId = Convert.ToInt32(Session["UserId"]);
   if (userId == 0)
   {
      return RedirectToAction("Login", "Regjistrohu");
   }
   string[] allowedFileTypes = { ".png", ".jpeg", ".jpg" };
   var file = obj.Picture;
   bool u_gjet = false;
   var extension = Path.GetExtension(file.FileName);
   User u = db.User.Find(userId);
   if (file != null)
   {
      for (int i = 0; i < 3; i++)
      {
         if (extension.Equals(allowedFileTypes[i]))
         {
            u_gjet = true;
         }
      }
      if (u_gjet == true)
      {
         //Update Image
         string id_and_extension = userId + extension;
         string imgUrl = "~/Profile Images/" + id_and_extension;
         u.ImageUrl = imgUrl;
         db.Entry(u).State = EntityState.Modified;
```

```
                db.SaveChanges();
                var path = Server.MapPath("~/Profile Images/");
                if (!Directory.Exists(path))
                {
                    Directory.CreateDirectory(path);
                }
                if ((System.IO.File.Exists(path + id_and_extension)))
                {
                    System.IO.File.Delete(path + id_and_extension);
                }
                file.SaveAs((path + id_and_extension));
                return RedirectToAction("UserProfile");
            }
            else
            {
                ViewBag.Photo = ("Ju duhet te zgjidhni nje imazh");
                return RedirectToAction("UserProfile");
            }


        }
        return RedirectToAction("UserProfile");

    }


  }
}
```

Register View

```
@model ELearningProject.ViewModel.RegisterVM
@{
    ViewBag.Title = "Register";
    Layout = "~/Views/Shared/_UserLayout.cshtml";
}

<h2 class="text-success">Register</h2>



@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })

        <div class="form-group">
            @Html.LabelFor(model => model.Emri, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Emri, new { htmlAttributes = new { @class = "form-control" }
})
                @Html.ValidationMessageFor(model => model.Emri, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
```

```
            @Html.LabelFor(model => model.Mbiemri, htmlAttributes: new { @class = "control-label col-md-
2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Mbiemri, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.Mbiemri, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Username, htmlAttributes: new { @class = "control-label col-
md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Username, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.Username, "", new { @class = "text-danger" })
                <p class="text-danger">@ViewBag.UserNameMessage</p>
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-
2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Password, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
```

```
            @Html.LabelFor(model => model.ConfirmPassword, htmlAttributes: new { @class = "control-label
col-md-2" })

            <div class="col-md-10">

                @Html.EditorFor(model => model.ConfirmPassword, new { htmlAttributes = new { @class =
"form-control" } })

                @Html.ValidationMessageFor(model => model.ConfirmPassword, "", new { @class = "text-
danger" })

            </div>

        </div>


        <div class="form-group">

            @Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "control-label col-md-2"
})

            <div class="col-md-10">

                @Html.EditorFor(model => model.Email, new { htmlAttributes = new { @class = "form-control"
} })

                @Html.ValidationMessageFor(model => model.Email, "", new { @class = "text-danger" })

                <p class="text-danger">@ViewBag.EmailMessage</p>

            </div>

        </div>


        <div class="form-group">

            <div class="col-md-offset-2 col-md-10">

                <input type="submit" value="Register" class="btn btn-primary" style="border-radius:20px" />

            </div>

        </div>

    </div>

}
```

```
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

LoginView

```
@model ELearningProject.ViewModel.LoginVM
@{
    ViewBag.Title = "Login";
    Layout = "~/Views/Shared/_UserLayout.cshtml";
}

<h2>Login</h2>



@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <p class="text-danger">@ViewBag.Error</p>
        <div class="form-group">
            @Html.LabelFor(model => model.Username, htmlAttributes: new { @class = "control-label col-
md-2" })
```

```
            <div class="col-md-10">
                @Html.EditorFor(model => model.Username, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.Username, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Password, htmlAttributes: new { @class = "control-label col-md-
2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Password, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.Password, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Login" class="btn btn-success" /><br /><br />
            </div>
        </div>
    </div>
}



@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

ChangePasswordView

```
@model ELearningProject.ViewModel.ChangePasswordVM
@{
    ViewBag.Title = "Change Password";
    Layout = "~/Views/Shared/_UserLayout.cshtml";
}



@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Change Password</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.OldPassword, htmlAttributes: new { @class = "control-label col-
md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.OldPassword, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.OldPassword, "", new { @class = "text-danger"
})
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.NewPassword, htmlAttributes: new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.NewPassword, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.NewPassword, "", new { @class = "text-danger"
})
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.ConfirmPassword, htmlAttributes: new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.ConfirmPassword, new { htmlAttributes = new { @class =
"form-control" } })
                @Html.ValidationMessageFor(model => model.ConfirmPassword, "", new { @class = "text-
danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Change Password" class="btn btn-default" />
            </div>
        </div>
    </div>
}
```

```
@section Scripts {
   @Scripts.Render("~/bundles/jqueryval")
}
```

UserProfileView

```
@model ELearningProject.Models.User
@{
   ViewBag.Title = "User Profile";
   Layout = "~/Views/Shared/_UserLayout.cshtml";
}

<h2 class="text-center">My Profile</h2>

<div class="well">
   <hr />
   <dl class="dl-horizontal">
      <dt>
         @Html.DisplayNameFor(model => model.Emri)
      </dt>

      <dd>
         @Html.DisplayFor(model => model.Emri)
      </dd>
      <dt>
         @Html.DisplayNameFor(model => model.Mbiemri)
```

```
</dt>

<dd>
    @Html.DisplayFor(model => model.Mbiemri)
</dd>
<dt>
    @Html.DisplayNameFor(model => model.Username)
</dt>

<dd>
    @Html.DisplayFor(model => model.Username)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.Password)
</dt>

<dd>
    @Html.DisplayFor(model => model.Password)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.Email)
</dt>

<dd>
    @Html.DisplayFor(model => model.Email)
</dd>
<dt>
```

```
            Account Created
        </dt>


        <dd>
            @Html.DisplayFor(model => model.CreatedOn)
        </dd>
        <dt>
            @Html.DisplayNameFor(model => model.ImageUrl)
        </dt>
        <dd style="margin-top:20px">
            @if (Model.ImageUrl.Length > 0)
            {
                <img src="@Url.Content(Model.ImageUrl)" alt="No Image" style="height:200px" class="img-
responsive img-circle" />
            }
            else
            {
                <img src="@Url.Content("~/Profile Images/WF.jpg")" alt="No Image" style="height:200px"
class="img-responsive img-circle" />
            }
            <br /><br />
            @using (Html.BeginForm("UpdatePicture", "User", FormMethod.Post, new { enctype =
"multipart/form-data" }))
            {
                <input type="file" name="Picture" class="col-sm-2 form-control" required /><br /><br />


                <input type="submit" value="Save Image" class="btn btn-default" />
                <p class="text-danger">@ViewBag.Photo</p>


            }
            <br />
            @Html.ActionLink("Ndrysho Password", "ChangePassword", "User")
        </dd>

    </dl>

</div>
```