6/30/2021

Web Engineering

Name:

Muhammad Haseeb Ahmad	BSE183010
Mohammad Abdul Rafay	BSE183009

Course: Web Engineering

Course Code: SE3523

Assignment: 04

Submitted to: Respected Sir Samir Obaid

Dated: 30th June, 2021

Web Engineering
SE3523

Table of Contents

Model:

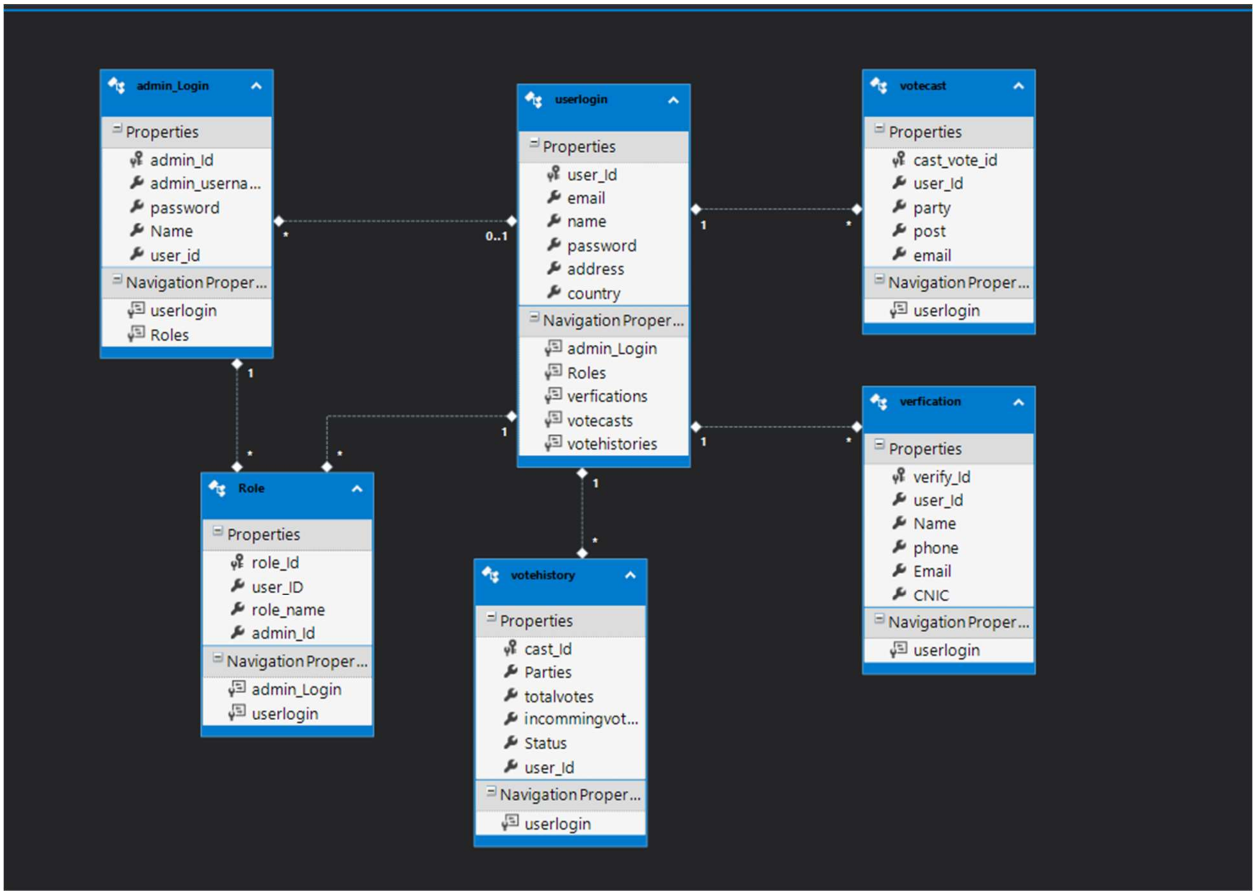


Figure 1: Model

Controllers:

Account Controller:

```
using System;

using System.Collections.Generic;

using System.Net.Http;

using System.Security.Claims;

using System.Security.Cryptography;

using System.Threading.Tasks;

using System.Web;

using System.Web.Http;

using System.Web.Http.ModelBinding;

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.OAuth;
using WebApplication2.Models;
using WebApplication2.Providers;
using WebApplication2.Results;

namespace WebApplication2.Controllers
{
    [Authorize]
    [RoutePrefix("api/Account")]
    public class AccountController : ApiController
    {
        private const string LocalLoginProvider = "Local";
        private ApplicationUserManager _userManager;

        public AccountController()
        {
        }

        public AccountController(ApplicationUserManager userManager,
            ISecureDataFormat<AuthenticationTicket> accessTokenFormat)
        {
            UserManager = userManager;
            AccessTokenFormat = accessTokenFormat;
        }
        public ApplicationUserManager UserManager
        {
            get
            {
                return _userManager ?? Request.GetOwinContext().GetUserManager<ApplicationUserManager>();
            }
            private set
            {
                _userManager = value;
            }
        }

        public ISecureDataFormat<AuthenticationTicket> AccessTokenFormat { get; private set; }

        // GET api/Account/UserInfo
        [HostAuthentication(DefaultAuthenticationTypes.ExternalBearer)]
        [Route("UserInfo")]
        public UserInfoViewModel GetUserInfo()
        {
            ExternalLoginData externalLogin = ExternalLoginData.FromIdentity(User.Identity as ClaimsIdentity);

            return new UserInfoViewModel
            {
                Email = User.Identity.GetUserName(),
                HasRegistered = externalLogin == null,
                LoginProvider = externalLogin != null ? externalLogin.LoginProvider : null
            };
```

```
}

// POST api/Account/Logout
[Route("Logout")]
public IHttpActionResult Logout()
{
    Authentication.SignOut(CookieAuthenticationDefaults.AuthenticationType);
    return Ok();
}

// GET api/Account/ManageInfo?returnUrl=%2F&generateState=true
[Route("ManageInfo")]
public async Task<ManageInfoViewModel> GetManageInfo(string returnUrl, bool generateState = false)
{
    IdentityUser user = await UserManager.FindByIdAsync(User.Identity.GetUserId());

    if (user == null)
    {
        return null;
    }

    List<UserLoginInfoViewModel> logins = new List<UserLoginInfoViewModel>();

    foreach (IdentityUserLogin linkedAccount in user.Logins)
    {
        logins.Add(new UserLoginInfoViewModel
        {
            LoginProvider = linkedAccount.LoginProvider,
            ProviderKey = linkedAccount.ProviderKey
        });
    }

    if (user.PasswordHash != null)
    {
        logins.Add(new UserLoginInfoViewModel
        {
            LoginProvider = LocalLoginProvider,
            ProviderKey = user.UserName,
        });
    }

    return new ManageInfoViewModel
    {
        LocalLoginProvider = LocalLoginProvider,
        Email = user.UserName,
        Logins = logins,
        ExternalLoginProviders = GetExternalLogins(returnUrl, generateState)
    };
}

// POST api/Account/ChangePassword
[Route("ChangePassword")]
public async Task<IHttpActionResult> ChangePassword(ChangePasswordBindingModel model)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    IdentityResult result = await UserManager.ChangePasswordAsync(User.Identity.GetUserId(),
model.OldPassword,
        model.NewPassword);

    if (!result.Succeeded)
    {
        return GetErrorResult(result);
    }

    return Ok();
}

// POST api/Account/SetPassword
[Route("SetPassword")]
public async Task<IHttpActionResult> SetPassword(SetPasswordBindingModel model)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    IdentityResult result = await UserManager.AddPasswordAsync(User.Identity.GetUserId(),
model.NewPassword);

    if (!result.Succeeded)
    {
        return GetErrorResult(result);
    }

    return Ok();
}

// POST api/Account/AddExternalLogin
[Route("AddExternalLogin")]
public async Task<IHttpActionResult> AddExternalLogin(AddExternalLoginBindingModel model)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    Authentication.SignOut(DefaultAuthenticationTypes.ExternalCookie);

    AuthenticationTicket ticket = AccessTokenFormat.Unprotect(model.ExternalAccessToken);

    if (ticket == null || ticket.Identity == null || (ticket.Properties != null
        && ticket.Properties.ExpiresUtc.HasValue
        && ticket.Properties.ExpiresUtc.Value < DateTimeOffset.UtcNow))
    {
        return BadRequest("External login failure.");
    }

    ExternalLoginData externalData = ExternalLoginData.FromIdentity(ticket.Identity);

    if (externalData == null)
    {
        return BadRequest("The external login is already associated with an account.");
    }

    IdentityResult result = await UserManager.AddLoginAsync(User.Identity.GetUserId(),
        new UserLoginInfo(externalData.LoginProvider, externalData.ProviderKey));

    if (!result.Succeeded)
    {
        return GetErrorResult(result);
    }

    return Ok();
}

// POST api/Account/RemoveLogin
[Route("RemoveLogin")]
public async Task<IHttpActionResult> RemoveLogin(RemoveLoginBindingModel model)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    IdentityResult result;

    if (model.LoginProvider == LocalLoginProvider)
    {
        result = await UserManager.RemovePasswordAsync(User.Identity.GetUserId());
    }
```

```
            else
            {
                result = await UserManager.RemoveLoginAsync(User.Identity.GetUserId(),
                    new UserLoginInfo(model.LoginProvider, model.ProviderKey));
            }

            if (!result.Succeeded)
            {
                return GetErrorResult(result);
            }

            return Ok();
        }

        // GET api/Account/ExternalLogin
        [OverrideAuthentication]
        [HostAuthentication(DefaultAuthenticationTypes.ExternalCookie)]
        [AllowAnonymous]
        [Route("ExternalLogin", Name = "ExternalLogin")]
        public async Task<IHttpActionResult> GetExternalLogin(string provider, string error = null)
        {
            if (error != null)
            {
                return Redirect(Url.Content("~/") + "#error=" + Uri.EscapeDataString(error));
            }

            if (!User.Identity.IsAuthenticated)
            {
                return new ChallengeResult(provider, this);
            }

            ExternalLoginData externalLogin = ExternalLoginData.FromIdentity(User.Identity as
    ClaimsIdentity);

            if (externalLogin == null)
            {
                return InternalServerError();
            }

            if (externalLogin.LoginProvider != provider)
            {
                Authentication.SignOut(DefaultAuthenticationTypes.ExternalCookie);
                return new ChallengeResult(provider, this);
            }

            ApplicationUser user = await UserManager.FindAsync(new
    UserLoginInfo(externalLogin.LoginProvider,
                externalLogin.ProviderKey));

            bool hasRegistered = user != null;

            if (hasRegistered)
            {
                Authentication.SignOut(DefaultAuthenticationTypes.ExternalCookie);

                 ClaimsIdentity oAuthIdentity = await user.GenerateUserIdentityAsync(UserManager,
                    OAuthDefaults.AuthenticationType);
                ClaimsIdentity cookieIdentity = await user.GenerateUserIdentityAsync(UserManager,
                    CookieAuthenticationDefaults.AuthenticationType);

                AuthenticationProperties properties =
ApplicationOAuthProvider.CreateProperties(user.UserName);
                Authentication.SignIn(properties, oAuthIdentity, cookieIdentity);
            }
            else
            {
                IEnumerable<Claim> claims = externalLogin.GetClaims();
                ClaimsIdentity identity = new ClaimsIdentity(claims, OAuthDefaults.AuthenticationType);
                Authentication.SignIn(identity);
            }

            return Ok();
        }

        // GET api/Account/ExternalLogins?returnUrl=%2F&generateState=true
        [AllowAnonymous]
        [Route("ExternalLogins")]
        public IEnumerable<ExternalLoginViewModel> GetExternalLogins(string returnUrl, bool
generateState = false)
        {
            IEnumerable<AuthenticationDescription> descriptions =
Authentication.GetExternalAuthenticationTypes();
            List<ExternalLoginViewModel> logins = new List<ExternalLoginViewModel>();

            string state;

            if (generateState)
            {
```

```
            const int strengthInBits = 256;
            state = RandomOAuthStateGenerator.Generate(strengthInBits);
        }
        else
        {
            state = null;
        }

        foreach (AuthenticationDescription description in descriptions)
        {
            ExternalLoginViewModel login = new ExternalLoginViewModel
            {
                Name = description.Caption,
                Url = Url.Route("ExternalLogin", new
                {
                    provider = description.AuthenticationType,
                    response_type = "token",
                    client_id = Startup.PublicClientId,
                    redirect_uri = new Uri(Request.RequestUri, returnUrl).AbsoluteUri,
                    state = state
                }),
                State = state
            };
            logins.Add(login);
        }

        return logins;
    }
```

```
// POST api/Account/Register
[AllowAnonymous]
[Route("Register")]
public async Task<IHttpActionResult> Register(RegisterBindingModel model)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    var user = new ApplicationUser() { UserName = model.Email, Email = model.Email };

    IdentityResult result = await UserManager.CreateAsync(user, model.Password);

    if (!result.Succeeded)
    {
        return GetErrorResult(result);
    }

    return Ok();
}

// POST api/Account/RegisterExternal
[OverrideAuthentication]
[HostAuthentication(DefaultAuthenticationTypes.ExternalBearer)]
[Route("RegisterExternal")]
public async Task<IHttpActionResult> RegisterExternal(RegisterExternalBindingModel model)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    var info = await Authentication.GetExternalLoginInfoAsync();
    if (info == null)
    {
        return InternalServerError();
    }

    var user = new ApplicationUser() { UserName = model.Email, Email = model.Email };

    IdentityResult result = await UserManager.CreateAsync(user);
    if (!result.Succeeded)
    {
        return GetErrorResult(result);
    }

    result = await UserManager.AddLoginAsync(user.Id, info.Login);
    if (!result.Succeeded)
    {
        return GetErrorResult(result);
    }
    return Ok();
}

protected override void Dispose(bool disposing)
{
    if (disposing && _userManager != null)
```

```
            {
                _userManager.Dispose();
                _userManager = null;
            }

            base.Dispose(disposing);
        }

        #region Helpers

        private IAuthenticationManager Authentication
        {
            get { return Request.GetOwinContext().Authentication; }
        }

        private IHttpActionResult GetErrorResult(IdentityResult result)
        {
            if (result == null)
            {
                return InternalServerError();
            }

            if (!result.Succeeded)
            {
                if (result.Errors != null)
                {
                    foreach (string error in result.Errors)
                    {
                        ModelState.AddModelError("", error);
                    }
                }

                if (ModelState.IsValid)
                {
                    // No ModelState errors are available to send, so just return an empty BadRequest.
                    return BadRequest();
                }

                return BadRequest(ModelState);
            }

            return null;
        }

        private class ExternalLoginData
        {
            public string LoginProvider { get; set; }
            public string ProviderKey { get; set; }
            public string UserName { get; set; }

            public IList<Claim> GetClaims()
            {
                IList<Claim> claims = new List<Claim>();
                claims.Add(new Claim(ClaimTypes.NameIdentifier, ProviderKey, null, LoginProvider));

                if (UserName != null)
                {
                    claims.Add(new Claim(ClaimTypes.Name, UserName, null, LoginProvider));
                }

                return claims;
            }

            public static ExternalLoginData FromIdentity(ClaimsIdentity identity)
            {
                if (identity == null)
                {
                    return null;
                }

                Claim providerKeyClaim = identity.FindFirst(ClaimTypes.NameIdentifier);

                if (providerKeyClaim == null || String.IsNullOrEmpty(providerKeyClaim.Issuer)
                    || String.IsNullOrEmpty(providerKeyClaim.Value))
                {
                    return null;
                }

                if (providerKeyClaim.Issuer == ClaimsIdentity.DefaultIssuer)
                {
                    return null;
                }

                return new ExternalLoginData
                {
                    LoginProvider = providerKeyClaim.Issuer,
                    ProviderKey = providerKeyClaim.Value,
                  UserName = identity.FindFirstValue(ClaimTypes.Name)
                };
              }
            }

            private static class RandomOAuthStateGenerator
            {
              private static RandomNumberGenerator _random = new RNGCryptoServiceProvider();

              public static string Generate(int strengthInBits)
              {
                const int bitsPerByte = 8;

                if (strengthInBits % bitsPerByte != 0)
                {
                  throw new ArgumentException("strengthInBits must be evenly divisible by 8.", "strengthInBits");
                }

                int strengthInBytes = strengthInBits / bitsPerByte;

                byte[] data = new byte[strengthInBytes];
                _random.GetBytes(data);
                return HttpServerUtility.UrlTokenEncode(data);
              }
            }

            #endregion
          }
```

```
}
```

Admin Controller:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Description;
using WebApplication2.Models;

namespace WebApplication2.Controllers
{
    public class admin_LoginController : ApiController
    {
        private voteDBEntities1 db = new voteDBEntities1();

        // GET: api/admin_Login
        public IQueryable<admin_Login> Getadmin_Login()
        {
            return db.admin_Login;
        }

        // GET: api/admin_Login/5
        [ResponseType(typeof(admin_Login))]
        public IHttpActionResult Getadmin_Login(int id)
        {
            admin_Login admin_Login = db.admin_Login.Find(id);
            if (admin_Login == null)
            {
                return NotFound();
            }

            return Ok(admin_Login);
        }

        // PUT: api/admin_Login/5
        [ResponseType(typeof(void))]
        public IHttpActionResult Putadmin_Login(int id, admin_Login admin_Login)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            if (id != admin_Login.admin_Id)
            {
                return BadRequest();
            }

            db.Entry(admin_Login).State = EntityState.Modified;

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!admin_LoginExists(id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }

            return StatusCode(HttpStatusCode.NoContent);
        }

        // POST: api/admin_Login
        [ResponseType(typeof(admin_Login))]
        public IHttpActionResult Postadmin_Login(admin_Login admin_Login)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            db.admin_Login.Add(admin_Login);

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateException)
            {
                if (admin_LoginExists(admin_Login.admin_Id))
                {
                    return Conflict();
                }
                else
                {
                    throw;
                }
            }

            return CreatedAtRoute("DefaultApi", new { id = admin_Login.admin_Id },
admin_Login);
        }

        // DELETE: api/admin_Login/5
        [ResponseType(typeof(admin_Login))]
        public IHttpActionResult Deleteadmin_Login(int id)
        {
            admin_Login admin_Login = db.admin_Login.Find(id);
            if (admin_Login == null)
            {
                return NotFound();
            }
            db.admin_Login.Remove(admin_Login);
            db.SaveChanges();

            return Ok(admin_Login);
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }

        private bool admin_LoginExists(int id)
        {
            return db.admin_Login.Count(e => e.admin_Id == id) > 0;
        }
    }
}
```

Home Controller:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace WebApplication2.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewBag.Title = "Home Page";
            /*
            using (DBEntities db = new DBEntities)
            {
                List<admin_LoginController> admin_Logins = db.Admin_login.ToList();
                List<userloginsController> user_Logins = db.userlogin.ToList();
                List<verficationsController> verfiy = db.verfication.ToList();
                List<votecastsController> votecasts = db.castvote.ToList();
                List<votehistoriesController> history = db.history.ToList();

                var q = (from pd in DataContext.userlogin
                         join od in DataContext.castvote on pd.userlogin equals
od.userlogin
                         orderby od.castvote
                         select new
                         {
                             od.user_Id,
                             pd.name,
                             pd.email,
                             od.password,
                             od.country,
                             pd.user_id,
                         }).ToList();
            }
            */
            return View();
        }
    }
}
```

Roles Controller:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Description;
using WebApplication2.Models;

namespace WebApplication2.Controllers
{
    public class RolesController : ApiController
    {
        private voteDBEntities1 db = new voteDBEntities1();

        // GET: api/Roles
        public IQueryable<Role> GetRoles()
        {
            return db.Roles;
        }

        // GET: api/Roles/5
        [ResponseType(typeof(Role))]
        public IHttpActionResult GetRole(int id)
        {
            Role role = db.Roles.Find(id);
            if (role == null)
            {
                return NotFound();
            }

            return Ok(role);
        }

        // PUT: api/Roles/5
        [ResponseType(typeof(void))]
        public IHttpActionResult PutRole(int id, Role role)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }
            if (id != role.role_Id)
            {
                return BadRequest();
            }

            db.Entry(role).State = EntityState.Modified;

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!RoleExists(id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }

            return StatusCode(HttpStatusCode.NoContent);
        }

        // POST: api/Roles
        [ResponseType(typeof(Role))]
        public IHttpActionResult PostRole(Role role)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            db.Roles.Add(role);

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateException)
            {
                if (RoleExists(role.role_Id))
                {
                    return Conflict();
                }
                else
                {
                    throw;
                }
            }

            return CreatedAtRoute("DefaultApi", new { id = role.role_Id }, role);
        }

        // DELETE: api/Roles/5
        [ResponseType(typeof(Role))]
        public IHttpActionResult DeleteRole(int id)
        {
            Role role = db.Roles.Find(id);
            if (role == null)
            {
                return NotFound();
            }

            db.Roles.Remove(role);
            db.SaveChanges();

            return Ok(role);
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }

        private bool RoleExists(int id)
        {
            return db.Roles.Count(e => e.role_Id == id) > 0;
        }
    }
}
```

User Login Controller:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Description;
using WebApplication2.Models;

namespace WebApplication2.Controllers
{
    public class userloginsController : ApiController
    {
        private voteDBEntities1 db = new voteDBEntities1();

        // GET: api/userlogins
        public IQueryable<userlogin> Getuserlogins()
        {
            return db.userlogins;
        }
        // GET: api/userlogins/5
        [ResponseType(typeof(userlogin))]
        public IHttpActionResult Getuserlogin(int id)
        {
            userlogin userlogin = db.userlogins.Find(id);
            if (userlogin == null)
            {
                return NotFound();
            }

            return Ok(userlogin);
        }

        // PUT: api/userlogins/5
        [ResponseType(typeof(void))]
        public IHttpActionResult Putuserlogin(int id, userlogin userlogin)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            if (id != userlogin.user_Id)
            {
                return BadRequest();
            }

            db.Entry(userlogin).State = EntityState.Modified;

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!userloginExists(id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }

            return StatusCode(HttpStatusCode.NoContent);
        }

        // POST: api/userlogins
        [ResponseType(typeof(userlogin))]
        public IHttpActionResult Postuserlogin(userlogin userlogin)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            db.userlogins.Add(userlogin);
            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateException)
            {
                if (userloginExists(userlogin.user_Id))
                {
                    return Conflict();
                }
                else
                {
                    throw;
                }
            }

            return CreatedAtRoute("DefaultApi", new { id = userlogin.user_Id },
userlogin);
        }

        // DELETE: api/userlogins/5
        [ResponseType(typeof(userlogin))]
        public IHttpActionResult Deleteuserlogin(int id)
        {
            userlogin userlogin = db.userlogins.Find(id);
            if (userlogin == null)
            {
                return NotFound();
            }

            db.userlogins.Remove(userlogin);
            db.SaveChanges();

            return Ok(userlogin);
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }

        private bool userloginExists(int id)
        {
            return db.userlogins.Count(e => e.user_Id == id) > 0;
        }
    }
}
```

Values Controller:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace WebApplication2.Controllers
{
    [Authorize]
    public class ValuesController : ApiController
    {
        // GET api/values
        public IEnumerable<string> Get()
        {
            return new string[] { "value1", "value2" };
        }

        // GET api/values/5
        public string Get(int id)
        {
            return "value";
        }

        // POST api/values
        public void Post([FromBody]string value)
        {
        }

        // PUT api/values/5
        public void Put(int id, [FromBody]string value)
        {
        }

        // DELETE api/values/5
        public void Delete(int id)
        {
        }
    }
}
```

Verification Controller:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Description;
using WebApplication2.Models;

namespace WebApplication2.Controllers
{
    public class verficationsController : ApiController
    {
```

```
private voteDBEntities1 db = new voteDBEntities1();

// GET: api/verfications
public IQueryable<verfication> Getverfications()
{
    return db.verfications;
}

// GET: api/verfications/5
[ResponseType(typeof(verfication))]
public IHttpActionResult Getverfication(int id)
{
    verfication verfication = db.verfications.Find(id);
    if (verfication == null)
    {
        return NotFound();
    }

    return Ok(verfication);
}

// PUT: api/verfications/5
[ResponseType(typeof(void))]
public IHttpActionResult Putverfication(int id, verfication verfication)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }

    if (id != verfication.verify_Id)
    {
        return BadRequest();
    }

    db.Entry(verfication).State = EntityState.Modified;

    try
    {
        db.SaveChanges();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!verficationExists(id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }

    return StatusCode(HttpStatusCode.NoContent);
}

// POST: api/verfications
[ResponseType(typeof(verfication))]
```

```
        public IHttpActionResult Postverfication(verfication verfication)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            db.verfications.Add(verfication);

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateException)
            {
                if (verficationExists(verfication.verify_Id))
                {
                    return Conflict();
                }
                else
                {
                    throw;
                }
            }

            return CreatedAtRoute("DefaultApi", new { id = verfication.verify_Id },
verfication);
        }

        // DELETE: api/verfications/5
        [ResponseType(typeof(verfication))]
        public IHttpActionResult Deleteverfication(int id)
        {
            verfication verfication = db.verfications.Find(id);
            if (verfication == null)
            {
                return NotFound();
            }

            db.verfications.Remove(verfication);
            db.SaveChanges();

            return Ok(verfication);
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }

        private bool verficationExists(int id)
        {
            return db.verfications.Count(e => e.verify_Id == id) > 0;
        }
```

```
    }
}
```

Vote Cast Controller:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Description;
using WebApplication2.Models;

namespace WebApplication2.Controllers
{
    public class votecastsController : ApiController
    {
        private voteDBEntities1 db = new voteDBEntities1();

        // GET: api/votecasts
        public IQueryable<votecast> Getvotecasts()
        {
            return db.votecasts;
        }

        // GET: api/votecasts/5
        [ResponseType(typeof(votecast))]
        public IHttpActionResult Getvotecast(int id)
        {
            votecast votecast = db.votecasts.Find(id);
            if (votecast == null)
            {
                return NotFound();
            }

            return Ok(votecast);
        }

        // PUT: api/votecasts/5
        [ResponseType(typeof(void))]
        public IHttpActionResult Putvotecast(int id, votecast votecast)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            if (id != votecast.cast_vote_id)
            {
                return BadRequest();
            }

            db.Entry(votecast).State = EntityState.Modified;
            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!votecastExists(id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }

            return StatusCode(HttpStatusCode.NoContent);
        }

        // POST: api/votecasts
        [ResponseType(typeof(votecast))]
        public IHttpActionResult Postvotecast(votecast votecast)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            db.votecasts.Add(votecast);

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateException)
            {
                if (votecastExists(votecast.cast_vote_id))
                {
                    return Conflict();
                }
                else
                {
                    throw;
                }
            }

            return CreatedAtRoute("DefaultApi", new { id = votecast.cast_vote_id },
votecast);
        }

        // DELETE: api/votecasts/5
        [ResponseType(typeof(votecast))]
        public IHttpActionResult Deletevotecast(int id)
        {
            votecast votecast = db.votecasts.Find(id);
            if (votecast == null)
            {
                return NotFound();
            }

            db.votecasts.Remove(votecast);
            db.SaveChanges();

            return Ok(votecast);
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }

        private bool votecastExists(int id)
        {
            return db.votecasts.Count(e => e.cast_vote_id == id) > 0;
        }
    }
}
```

Vote History Controller:

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Http.Description;
using WebApplication2.Models;

namespace WebApplication2.Controllers
{
    public class votehistoriesController : ApiController
    {
        private voteDBEntities1 db = new voteDBEntities1();

        // GET: api/votehistories
        public IQueryable<votehistory> Getvotehistories()
        {
            return db.votehistories;
        }

        // GET: api/votehistories/5
        [ResponseType(typeof(votehistory))]
        public IHttpActionResult Getvotehistory(int id)
        {
            votehistory votehistory = db.votehistories.Find(id);
            if (votehistory == null)
            {
                return NotFound();
            }

            return Ok(votehistory);
        }

        // PUT: api/votehistories/5
        [ResponseType(typeof(void))]
        public IHttpActionResult Putvotehistory(int id, votehistory votehistory)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            if (id != votehistory.cast_Id)
            {
                return BadRequest();
            }

            db.Entry(votehistory).State = EntityState.Modified;

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateConcurrencyException)
            {
                if (!votehistoryExists(id))
                {
                    return NotFound();
                }
                else
                {
                    throw;
                }
            }

            return StatusCode(HttpStatusCode.NoContent);
        }

        // POST: api/votehistories
        [ResponseType(typeof(votehistory))]
        public IHttpActionResult Postvotehistory(votehistory votehistory)
        {
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            db.votehistories.Add(votehistory);

            try
            {
                db.SaveChanges();
            }
            catch (DbUpdateException)
            {
                if (votehistoryExists(votehistory.cast_Id))
                {
                    return Conflict();
                }
                else
                {
                    throw;
                }
            }

            return CreatedAtRoute("DefaultApi", new { id = votehistory.cast_Id }, 
votehistory);
        }

        // DELETE: api/votehistories/5
        [ResponseType(typeof(votehistory))]
        public IHttpActionResult Deletevotehistory(int id)
        {
            votehistory votehistory = db.votehistories.Find(id);
            if (votehistory == null)
            {
                return NotFound();
            }

            db.votehistories.Remove(votehistory);
            db.SaveChanges();

            return Ok(votehistory);
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }

        private bool votehistoryExists(int id)
        {
            return db.votehistories.Count(e => e.cast_Id == id) > 0;
        }
    }
}
```

App Start File:

BundleConfig.cs

```
using System.Web;
using System.Web.Optimization;

namespace WebApplication2
{
    public class BundleConfig
    {
        // For more information on bundling, visit
https://go.microsoft.com/fwlink/?LinkId=301862
        public static void RegisterBundles(BundleCollection bundles)
        {
            bundles.Add(new ScriptBundle("~/bundles/jquery").Include(
                        "~/Scripts/jquery-{version}.js"));

            // Use the development version of Modernizr to develop with and learn from.
Then, when you're
            // ready for production, use the build tool at https://modernizr.com to pick
only the tests you need.
            bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
                        "~/Scripts/modernizr-*"));

            bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(
                      "~/Scripts/bootstrap.js"));

            bundles.Add(new StyleBundle("~/Content/css").Include(
                      "~/Content/bootstrap.css",
                      "~/Content/site.css"));
        }
    }
}
```

FilterConfig.cs

```
using System.Web;
using System.Web.Mvc;

namespace WebApplication2
{
    public class FilterConfig
    {
        public static void RegisterGlobalFilters(GlobalFilterCollection filters)
        {
            filters.Add(new HandleErrorAttribute());
        }
    }
}
```

IdentityConfig.cs

```
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin;
using WebApplication2.Models;

namespace WebApplication2
{
    // Configure the application user manager used in this application. UserManager is
defined in ASP.NET Identity and is used by the application.

    public class ApplicationUserManager : UserManager<ApplicationUser>
    {
        public ApplicationUserManager(IUserStore<ApplicationUser> store)
            : base(store)
        {
        }

        public static ApplicationUserManager
Create(IdentityFactoryOptions<ApplicationUserManager> options, IOwinContext context)
        {
            var manager = new ApplicationUserManager(new
UserStore<ApplicationUser>(context.Get<ApplicationDbContext>()));
            // Configure validation logic for usernames
            manager.UserValidator = new UserValidator<ApplicationUser>(manager)
            {
                AllowOnlyAlphanumericUserNames = false,
                RequireUniqueEmail = true
            };
            // Configure validation logic for passwords
            manager.PasswordValidator = new PasswordValidator
            {
                RequiredLength = 6,
                RequireNonLetterOrDigit = true,
                RequireDigit = true,
                RequireLowercase = true,
                RequireUppercase = true,
            };
            var dataProtectionProvider = options.DataProtectionProvider;
            if (dataProtectionProvider != null)
            {
                manager.UserTokenProvider = new
DataProtectorTokenProvider<ApplicationUser>(dataProtectionProvider.Create("ASP.NET
Identity"));
            }
            return manager;
        }
    }
}
```

RouteConfig.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace WebApplication2
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id =
UrlParameter.Optional }
            );
        }
    }
}
```

Startup.Auth.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.Google;
using Microsoft.Owin.Security.OAuth;
using Owin;
using WebApplication2.Providers;
using WebApplication2.Models;

namespace WebApplication2
{
    public partial class Startup
    {
        public static OAuthAuthorizationServerOptions OAuthOptions { get; private set; }

        public static string PublicClientId { get; private set; }

        // For more information on configuring authentication, please visit
https://go.microsoft.com/fwlink/?LinkId=301864
        public void ConfigureAuth(IAppBuilder app)
        {
            // Configure the db context and user manager to use a single instance per
request
            app.CreatePerOwinContext(ApplicationDbContext.Create);

app.CreatePerOwinContext<ApplicationUserManager>(ApplicationUserManager.Create);

            // Enable the application to use a cookie to store information for the signed
in user
            // and to use a cookie to temporarily store information about a user logging
in with a third party login provider
            app.UseCookieAuthentication(new CookieAuthenticationOptions());
            app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);

            // Configure the application for OAuth based flow
            PublicClientId = "self";
            OAuthOptions = new OAuthAuthorizationServerOptions
            {
                TokenEndpointPath = new PathString("/Token"),
                Provider = new ApplicationOAuthProvider(PublicClientId),
                AuthorizeEndpointPath = new PathString("/api/Account/ExternalLogin"),
                AccessTokenExpireTimeSpan = TimeSpan.FromDays(14),
                // In production mode set AllowInsecureHttp = false
                AllowInsecureHttp = true
            };

            // Enable the application to use bearer tokens to authenticate users
            app.UseOAuthBearerTokens(OAuthOptions);

            // Uncomment the following lines to enable logging in with third party login
providers
            //app.UseMicrosoftAccountAuthentication(
            //    clientId: "",
            //    clientSecret: "");

            //app.UseTwitterAuthentication(
            //    consumerKey: "",
            //    consumerSecret: "");

            //app.UseFacebookAuthentication(
            //    appId: "",
            //    appSecret: "");

            //app.UseGoogleAuthentication(new GoogleOAuth2AuthenticationOptions()
            //{
            //    ClientId = "",
            //    ClientSecret = ""
            //});
        }
    }
}
```

WebApiConfig.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Web.Http;
using Microsoft.Owin.Security.OAuth;
using Newtonsoft.Json.Serialization;

namespace WebApplication2
{
    public static class WebApiConfig
    {
        public static void Register(HttpConfiguration config)
        {
            // Web API configuration and services
            // Configure Web API to use only bearer token authentication.
            config.SuppressDefaultHostAuthentication();
            config.Filters.Add(new
HostAuthenticationFilter(OAuthDefaults.AuthenticationType));

            // Web API routes
            config.MapHttpAttributeRoutes();
            config.Routes.MapHttpRoute(
                name: "DefaultApi",
                routeTemplate: "api/{controller}/{id}",
                defaults: new { id = RouteParameter.Optional }
            );
        }
    }
}
```

Models:

AccountBindingModels.cs

```
using System;
using System.ComponentModel.DataAnnotations;
using Newtonsoft.Json;

namespace WebApplication2.Models
{
    // Models used as parameters to AccountController actions.

    public class AddExternalLoginBindingModel
    {
        [Required]
        [Display(Name = "External access token")]
        public string ExternalAccessToken { get; set; }
    }

    public class ChangePasswordBindingModel
    {
        [Required]
        [DataType(DataType.Password)]
        [Display(Name = "Current password")]
        public string OldPassword { get; set; }

        [Required]
        [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters 
long.", MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "New password")]
        public string NewPassword { get; set; }

        [DataType(DataType.Password)]
        [Display(Name = "Confirm new password")]
        [Compare("NewPassword", ErrorMessage = "The new password and confirmation 
password do not match.")]
        public string ConfirmPassword { get; set; }
    }

    public class RegisterBindingModel
    {
        [Required]
        [Display(Name = "Email")]
        public string Email { get; set; }
        [Required]
        [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters 
long.", MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "Password")]
        public string Password { get; set; }

        [DataType(DataType.Password)]
        [Display(Name = "Confirm password")]
        [Compare("Password", ErrorMessage = "The password and confirmation password do 
not match.")]
        public string ConfirmPassword { get; set; }
    }

    public class RegisterExternalBindingModel
    {
        [Required]
        [Display(Name = "Email")]
        public string Email { get; set; }
    }

    public class RemoveLoginBindingModel
    {
        [Required]
        [Display(Name = "Login provider")]
        public string LoginProvider { get; set; }

        [Required]
        [Display(Name = "Provider key")]
        public string ProviderKey { get; set; }
    }

    public class SetPasswordBindingModel
    {
        [Required]
        [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters 
long.", MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "New password")]
        public string NewPassword { get; set; }

        [DataType(DataType.Password)]
        [Display(Name = "Confirm new password")]
        [Compare("NewPassword", ErrorMessage = "The new password and confirmation 
password do not match.")]
        public string ConfirmPassword { get; set; }
    }
}
```

AccountViewModels.cs

```
using System;
using System.Collections.Generic;
```

```
namespace WebApplication2.Models
{
    // Models returned by AccountController actions.

    public class ExternalLoginViewModel
    {
        public string Name { get; set; }

        public string Url { get; set; }

        public string State { get; set; }
    }

    public class ManageInfoViewModel
    {
        public string LocalLoginProvider { get; set; }

        public string Email { get; set; }

        public IEnumerable<UserLoginInfoViewModel> Logins { get; set; }

        public IEnumerable<ExternalLoginViewModel> ExternalLoginProviders { get; set; }
    }

    public class UserInfoViewModel
    {
        public string Email { get; set; }

        public bool HasRegistered { get; set; }

        public string LoginProvider { get; set; }
    }

    public class UserLoginInfoViewModel
    {
        public string LoginProvider { get; set; }

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

IdentityModels.cs

```
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;

namespace WebApplication2.Models
{
    // You can add profile data for the user by adding more properties to your
ApplicationUser class, please visit https://go.microsoft.com/fwlink/?LinkID=317594 to
learn more.
    public class ApplicationUser : IdentityUser
    {
        public async Task<ClaimsIdentity>
GenerateUserIdentityAsync(UserManager<ApplicationUser> manager, string
authenticationType)
        {
            // Note the authenticationType must match the one defined in
CookieAuthenticationOptions.AuthenticationType
            var userIdentity = await manager.CreateIdentityAsync(this,
authenticationType);
            // Add custom user claims here
            return userIdentity;
        }
    }

    public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
    {
        public ApplicationDbContext()
            : base("DefaultConnection", throwIfV1Schema: false)
        {
        }

        public static ApplicationDbContext Create()
        {
            return new ApplicationDbContext();
        }
    }
}
```

Providers:

ApplicationOAuthProvider.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.OAuth;
using WebApplication2.Models;

namespace WebApplication2.Providers
{
    public class ApplicationOAuthProvider : OAuthAuthorizationServerProvider
    {
        private readonly string _publicClientId;

        public ApplicationOAuthProvider(string publicClientId)
        {
            if (publicClientId == null)
            {
                throw new ArgumentNullException("publicClientId");
            }

            _publicClientId = publicClientId;
        }

        public override async Task
GrantResourceOwnerCredentials(OAuthGrantResourceOwnerCredentialsContext context)
        {
            var userManager =
context.OwinContext.GetUserManager<ApplicationUserManager>();

            ApplicationUser user = await userManager.FindAsync(context.UserName,
context.Password);

            if (user == null)
            {
                context.SetError("invalid_grant", "The user name or password is
incorrect.");
                return;
            }

            ClaimsIdentity oAuthIdentity = await
user.GenerateUserIdentityAsync(userManager,
               OAuthDefaults.AuthenticationType);
            ClaimsIdentity cookiesIdentity = await
user.GenerateUserIdentityAsync(userManager,
                CookieAuthenticationDefaults.AuthenticationType);

            AuthenticationProperties properties = CreateProperties(user.UserName);
            AuthenticationTicket ticket = new AuthenticationTicket(oAuthIdentity,
properties);
            context.Validated(ticket);
            context.Request.Context.Authentication.SignIn(cookiesIdentity);
        }

        public override Task TokenEndpoint(OAuthTokenEndpointContext context)
        {
            foreach (KeyValuePair<string, string> property in
context.Properties.Dictionary)
            {
                context.AdditionalResponseParameters.Add(property.Key, property.Value);
            }

            return Task.FromResult<object>(null);
        }

        public override Task
ValidateClientAuthentication(OAuthValidateClientAuthenticationContext context)
        {
            // Resource owner password credentials does not provide a client ID.
            if (context.ClientId == null)
            {
                context.Validated();
            }

            return Task.FromResult<object>(null);
        }
        public override Task
ValidateClientRedirectUri(OAuthValidateClientRedirectUriContext context)
        {
            if (context.ClientId == _publicClientId)
            {
                Uri expectedRootUri = new Uri(context.Request.Uri, "/");

                if (expectedRootUri.AbsoluteUri == context.RedirectUri)
                {
                    context.Validated();
                }
            }

            return Task.FromResult<object>(null);
        }

        public static AuthenticationProperties CreateProperties(string userName)
        {
            IDictionary<string, string> data = new Dictionary<string, string>
            {
                { "userName", userName }
            };
            return new AuthenticationProperties(data);
        }
    }
}
```

Results:

ChallengeResult.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Threading;
using System.Threading.Tasks;
using System.Web.Http;

namespace WebApplication2.Results
{
    public class ChallengeResult : IHttpActionResult
    {
        public ChallengeResult(string loginProvider, ApiController controller)
        {
            LoginProvider = loginProvider;
            Request = controller.Request;
        }

        public string LoginProvider { get; set; }
        public HttpRequestMessage Request { get; set; }
        public Task<HttpResponseMessage> ExecuteAsync(CancellationToken
cancellationToken)
        {
            Request.GetOwinContext().Authentication.Challenge(LoginProvider);

            HttpResponseMessage response = new
HttpResponseMessage(HttpStatusCode.Unauthorized);
            response.RequestMessage = Request;
            return Task.FromResult(response);
        }
    }
}
```

Databases Tables:

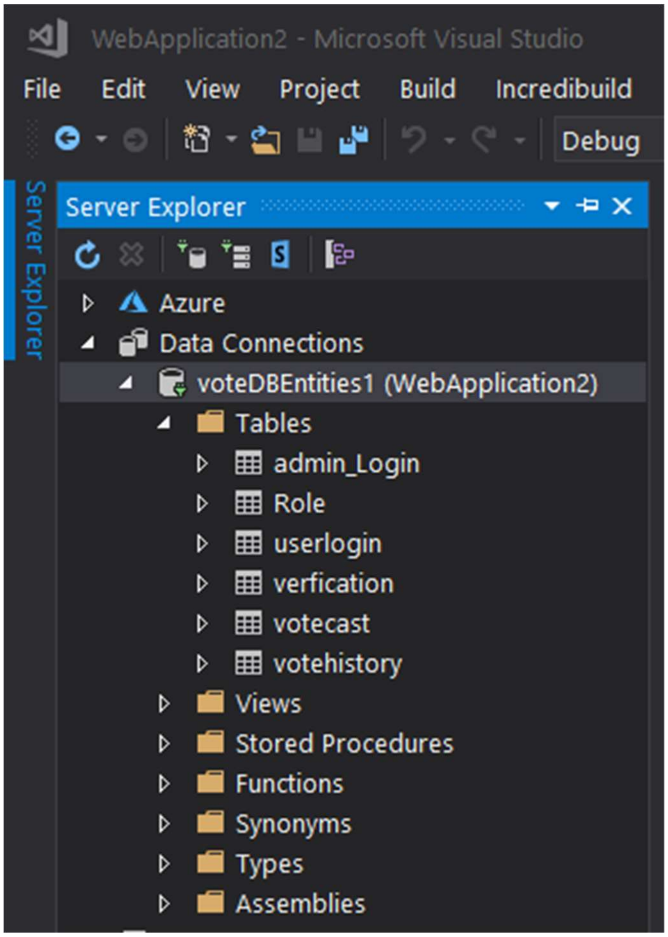
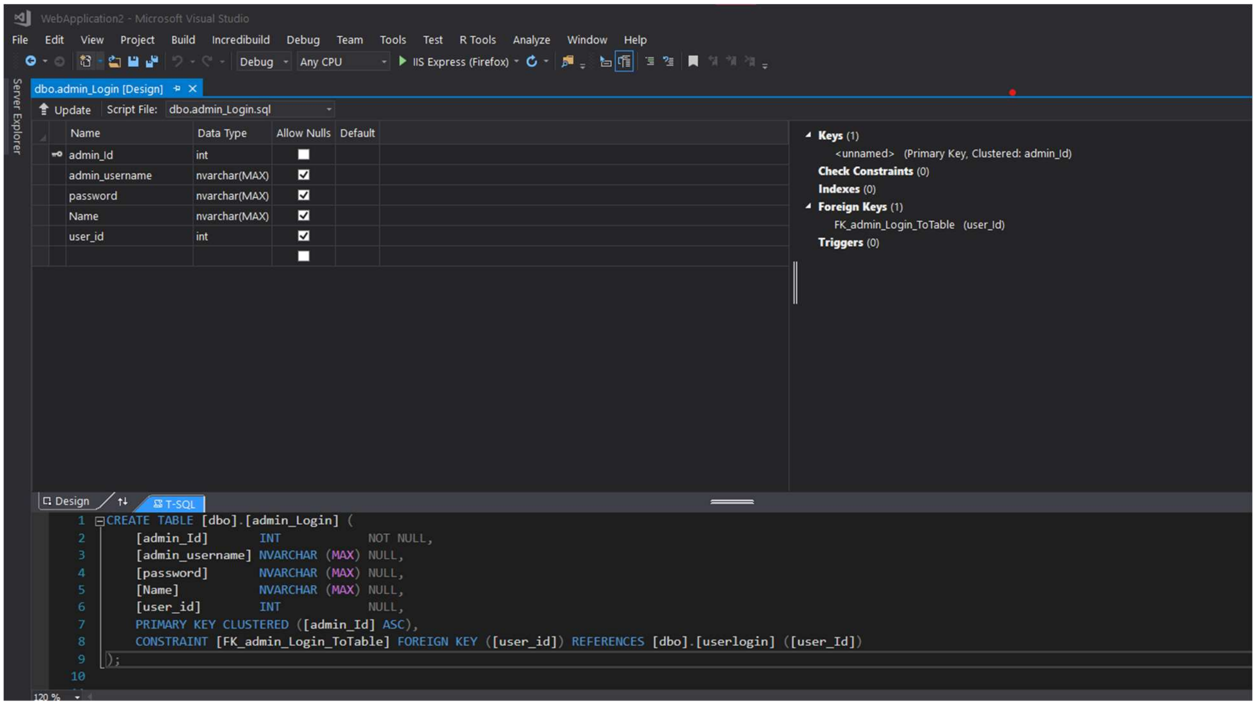


Figure 2:Tables

Tables:

Admin_login:

```
CREATE TABLE [dbo].[admin_Login] (
    [admin_Id]       INT            NOT NULL,
    [admin_username] NVARCHAR (MAX) NULL,
    [password]       NVARCHAR (MAX) NULL,
    [Name]           NVARCHAR (MAX) NULL,
    [user_id]        INT            NULL,
    PRIMARY KEY CLUSTERED ([admin_Id] ASC),
    CONSTRAINT [FK_admin_Login_ToTable] FOREIGN KEY ([user_id]) REFERENCES
[dbo].[userlogin] ([user_Id])
```



Script File: dbo.admin_Login.sql

	Name	Data Type	Allow Nulls	Default
🔑	admin_Id	int	□	
	admin_username	nvarchar(MAX)	☑	
	password	nvarchar(MAX)	☑	
	Name	nvarchar(MAX)	☑	
	user_id	int	☑	
			□	

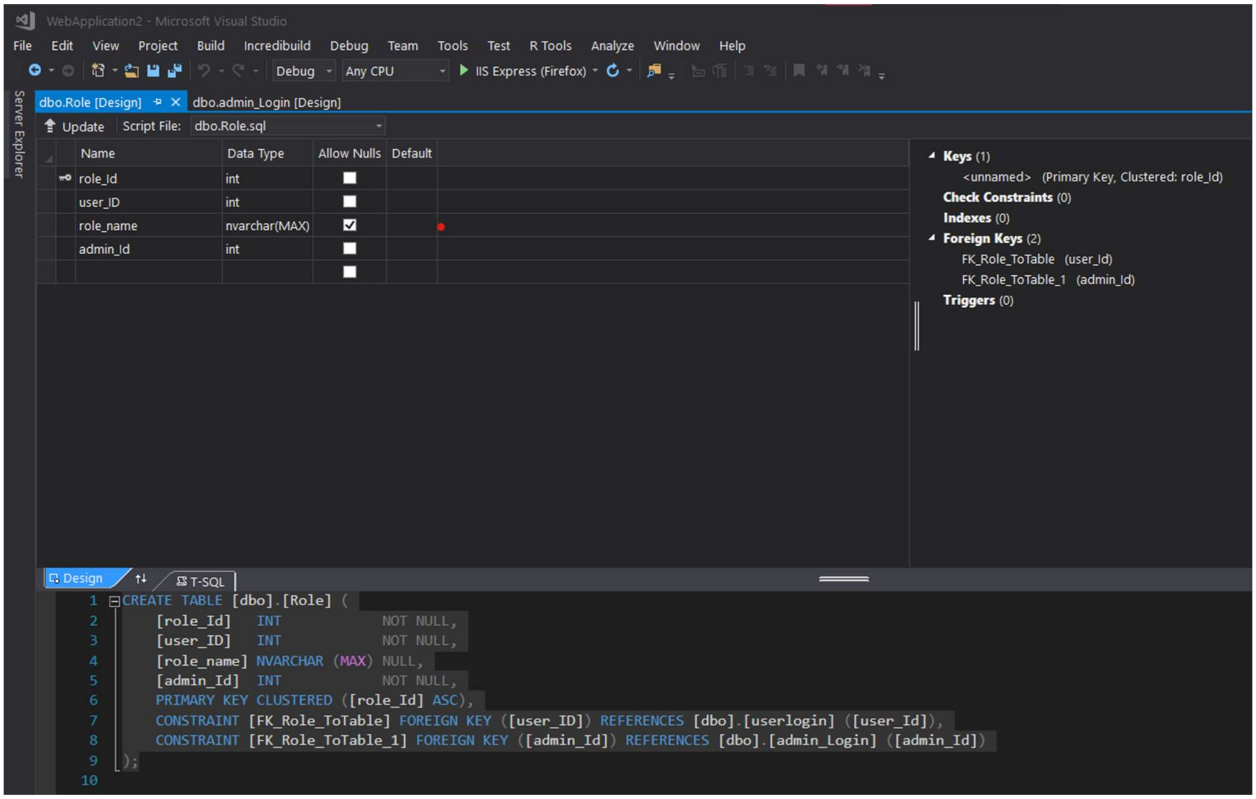
Keys (1)
<unnamed> (Primary Key, Clustered: admin_Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
FK_admin_Login_ToTable (user_Id)
Triggers (0)

```
CREATE TABLE [dbo].[admin_Login] (
    [admin_Id]       INT            NOT NULL,
    [admin_username] NVARCHAR (MAX) NULL,
    [password]       NVARCHAR (MAX) NULL,
    [Name]           NVARCHAR (MAX) NULL,
    [user_id]        INT            NULL,
    PRIMARY KEY CLUSTERED ([admin_Id] ASC),
    CONSTRAINT [FK_admin_Login_ToTable] FOREIGN KEY ([user_id]) REFERENCES [dbo].[userlogin] ([user_Id])
);
```

Figure 3: Admin login table

Role Table:

```
CREATE TABLE [dbo].[Role] (
    [role_Id]   INT            NOT NULL,
    [user_ID]   INT            NOT NULL,
    [role_name] NVARCHAR (MAX) NULL,
    [admin_Id]  INT            NOT NULL,
    PRIMARY KEY CLUSTERED ([role_Id] ASC),
    CONSTRAINT [FK_Role_ToTable] FOREIGN KEY ([user_ID]) REFERENCES [dbo].[userlogin]
([user_Id]),
    CONSTRAINT [FK_Role_ToTable_1] FOREIGN KEY ([admin_Id]) REFERENCES
[dbo].[admin_Login] ([admin_Id])
);
```


WebApplication2 - Microsoft Visual Studio
File Edit View Project Build Incredibuild Debug Team Tools Test R Tools Analyze Window Help
Debug Any CPU IIS Express (Firefox)
Server Explorer
dbo.Role [Design]
dbo.admin_Login [Design]
Update
Design
T-SQL

Script File: dbo.Role.sql

Name	Data Type	Allow Nulls	Default
role_Id	int	☐	
user_ID	int	☐	
role_name	nvarchar(MAX)	☑	
admin_Id	int	☐	
		☐	

Keys (1)
<unnamed> (Primary Key, Clustered: role_Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (2)
FK_Role_ToTable (user_Id)
FK_Role_ToTable_1 (admin_Id)
Triggers (0)

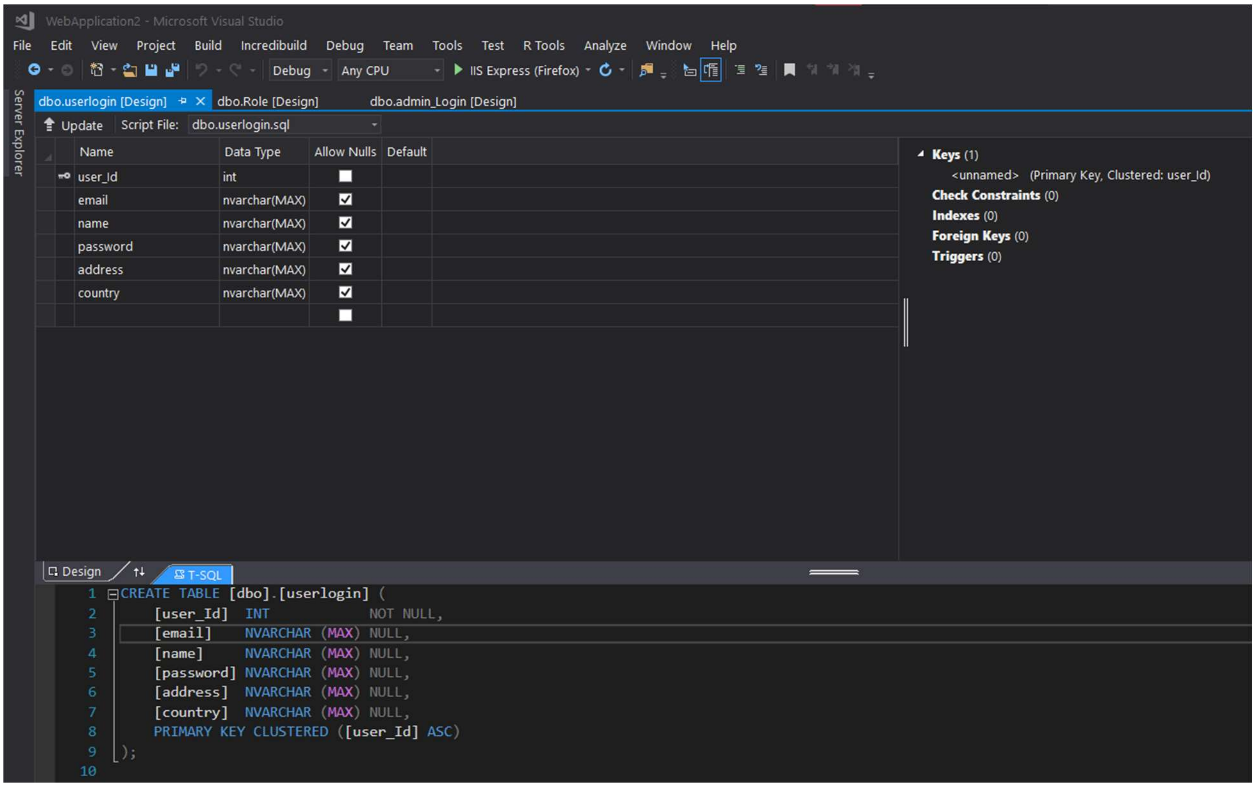
```sql
CREATE TABLE [dbo].[Role] (
    [role_Id]   INT            NOT NULL,
    [user_ID]   INT            NOT NULL,
    [role_name] NVARCHAR (MAX) NULL,
    [admin_Id]  INT            NOT NULL,
    PRIMARY KEY CLUSTERED ([role_Id] ASC),
    CONSTRAINT [FK_Role_ToTable] FOREIGN KEY ([user_ID]) REFERENCES [dbo].[userlogin] ([user_Id]),
    CONSTRAINT [FK_Role_ToTable_1] FOREIGN KEY ([admin_Id]) REFERENCES [dbo].[admin_Login] ([admin_Id])
);

```

Figure 4: Role Table

User Login:

```sql
CREATE TABLE [dbo].[userlogin] (
    [user_Id]  INT            NOT NULL,
    [email]    NVARCHAR (MAX) NULL,
    [name]     NVARCHAR (MAX) NULL,
    [password] NVARCHAR (MAX) NULL,
    [address]  NVARCHAR (MAX) NULL,
    [country]  NVARCHAR (MAX) NULL,
    PRIMARY KEY CLUSTERED ([user_Id] ASC)
);
```


WebApplication2 - Microsoft Visual Studio
File Edit View Project Build Incredibuild Debug Team Tools Test R Tools Analyze Window Help
Debug Any CPU IIS Express (Firefox)
Server Explorer
dbo.userlogin [Design]
dbo.Role [Design]
dbo.admin_Login [Design]
Update
Design
T-SQL

Script File: dbo.userlogin.sql

Name	Data Type	Allow Nulls	Default
user_Id	int	□	
email	nvarchar(MAX)	☑	
name	nvarchar(MAX)	☑	
password	nvarchar(MAX)	☑	
address	nvarchar(MAX)	☑	
country	nvarchar(MAX)	☑	
		□	

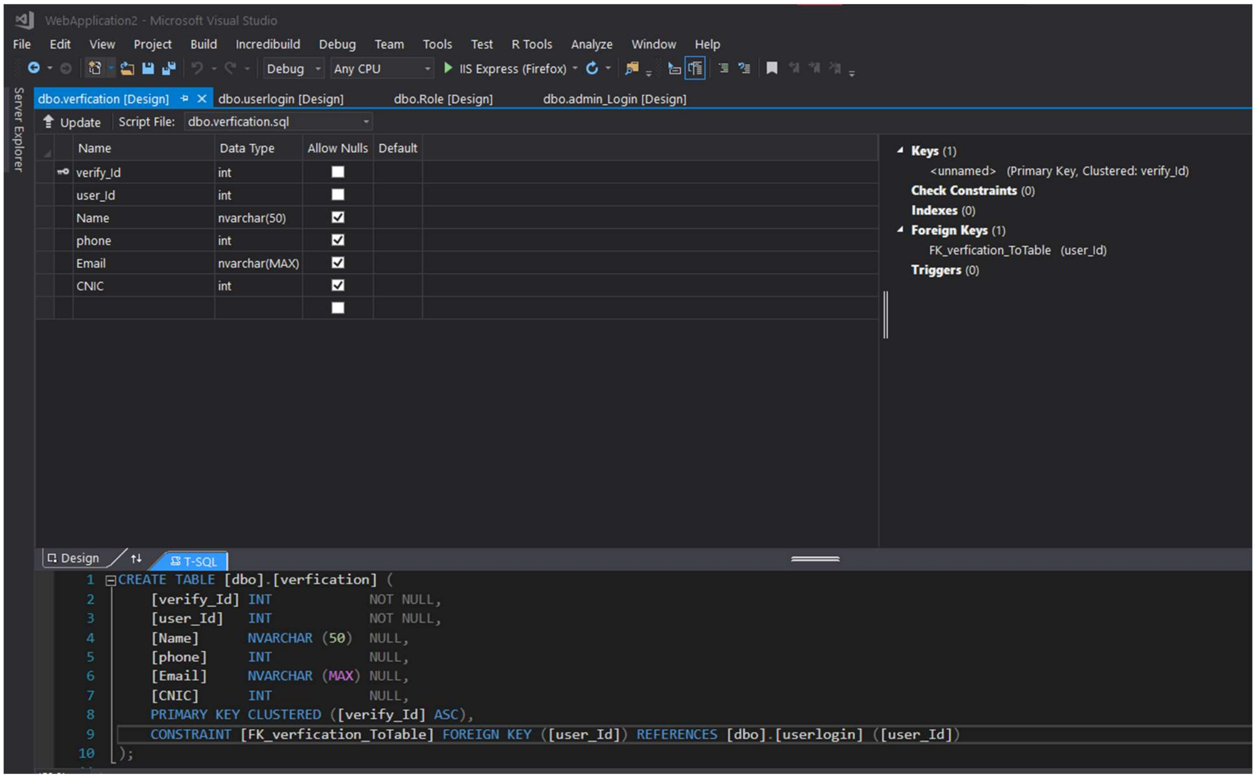
Keys (1)
<unnamed> (Primary Key, Clustered: user_Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

```
CREATE TABLE [dbo].[userlogin] (
    [user_Id]  INT            NOT NULL,
    [email]    NVARCHAR (MAX) NULL,
    [name]     NVARCHAR (MAX) NULL,
    [password] NVARCHAR (MAX) NULL,
    [address]  NVARCHAR (MAX) NULL,
    [country]  NVARCHAR (MAX) NULL,
    PRIMARY KEY CLUSTERED ([user_Id] ASC)
);
```

Figure 5: User Login Table

Verification Table:

```
CREATE TABLE [dbo].[verfication] (
    [verify_Id] INT            NOT NULL,
    [user_Id]   INT            NOT NULL,
    [Name]      NVARCHAR (50)  NULL,
    [phone]     INT            NULL,
    [Email]     NVARCHAR (MAX) NULL,
    [CNIC]      INT            NULL,
    PRIMARY KEY CLUSTERED ([verify_Id] ASC),
    CONSTRAINT [FK_verfication_ToTable] FOREIGN KEY ([user_Id]) REFERENCES
[dbo].[userlogin] ([user_Id])
);
```


WebApplication2 - Microsoft Visual Studio
File Edit View Project Build Incredibuild Debug Team Tools Test R Tools Analyze Window Help
Debug
Any CPU
IIS Express (Firefox)
dbo.verfication [Design]
dbo.userlogin [Design]
dbo.Role [Design]
dbo.admin_Login [Design]
Server Explorer
Update
Script File: dbo.verfication.sql
Design
T-SQL

Name	Data Type	Allow Nulls	Default
verify_Id	int	□	
user_Id	int	□	
Name	nvarchar(50)	☑	
phone	int	☑	
Email	nvarchar(MAX)	☑	
CNIC	int	☑	
		□	

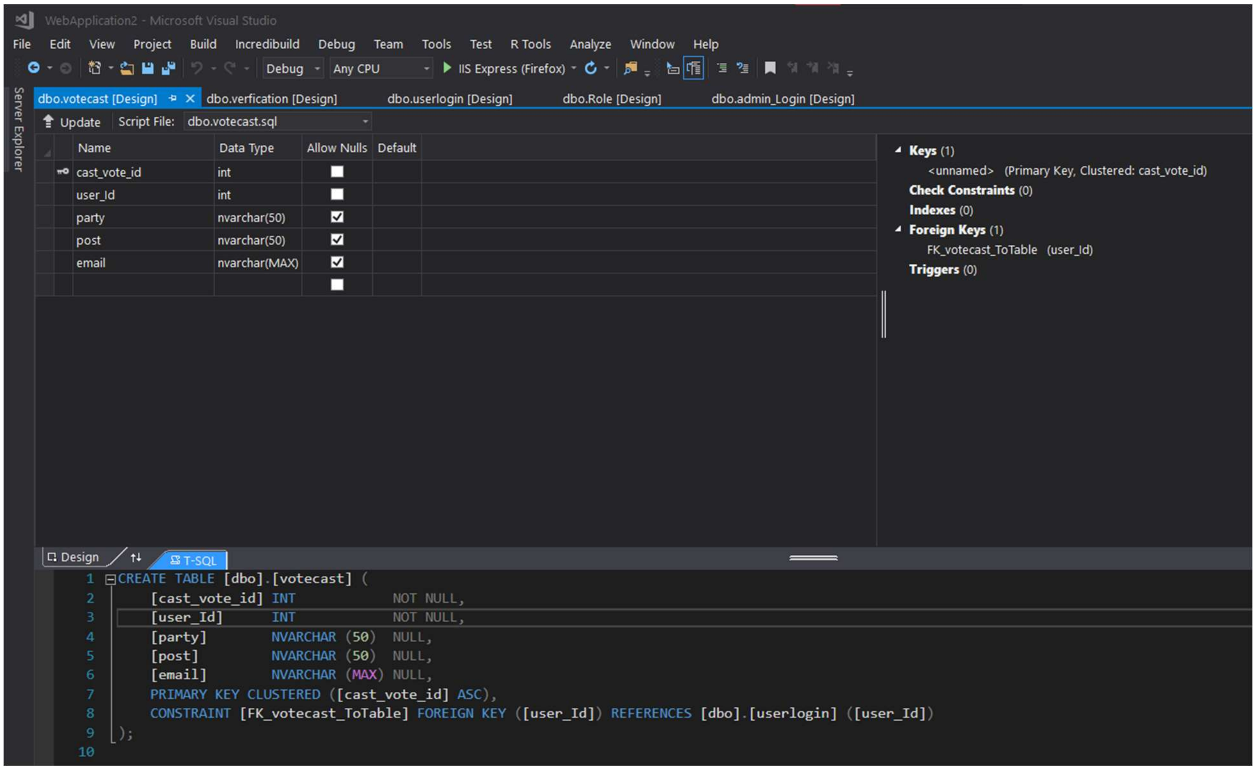
Keys (1)
<unnamed> (Primary Key, Clustered: verify_Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
FK_verfication_ToTable (user_Id)
Triggers (0)

```
CREATE TABLE [dbo].[verfication] (
    [verify_Id] INT            NOT NULL,
    [user_Id]   INT            NOT NULL,
    [Name]      NVARCHAR (50)  NULL,
    [phone]     INT            NULL,
    [Email]     NVARCHAR (MAX) NULL,
    [CNIC]      INT            NULL,
    PRIMARY KEY CLUSTERED ([verify_Id] ASC),
    CONSTRAINT [FK_verfication_ToTable] FOREIGN KEY ([user_Id]) REFERENCES [dbo].[userlogin] ([user_Id])
);
```

Figure 6: Verification Table

Vote Cast Table:

```
CREATE TABLE [dbo].[votecast] (
    [cast_vote_id] INT            NOT NULL,
    [user_Id]      INT            NOT NULL,
    [party]        NVARCHAR (50)  NULL,
    [post]         NVARCHAR (50)  NULL,
    [email]        NVARCHAR (MAX) NULL,
    PRIMARY KEY CLUSTERED ([cast_vote_id] ASC),
    CONSTRAINT [FK_votecast_ToTable] FOREIGN KEY ([user_Id]) REFERENCES [dbo].[userlogin]
([user_Id])
);
```



Name	Data Type	Allow Nulls	Default
cast_vote_id	int	□	
user_Id	int	□	
party	nvarchar(50)	☑	
post	nvarchar(50)	☑	
email	nvarchar(MAX)	☑	
		□	

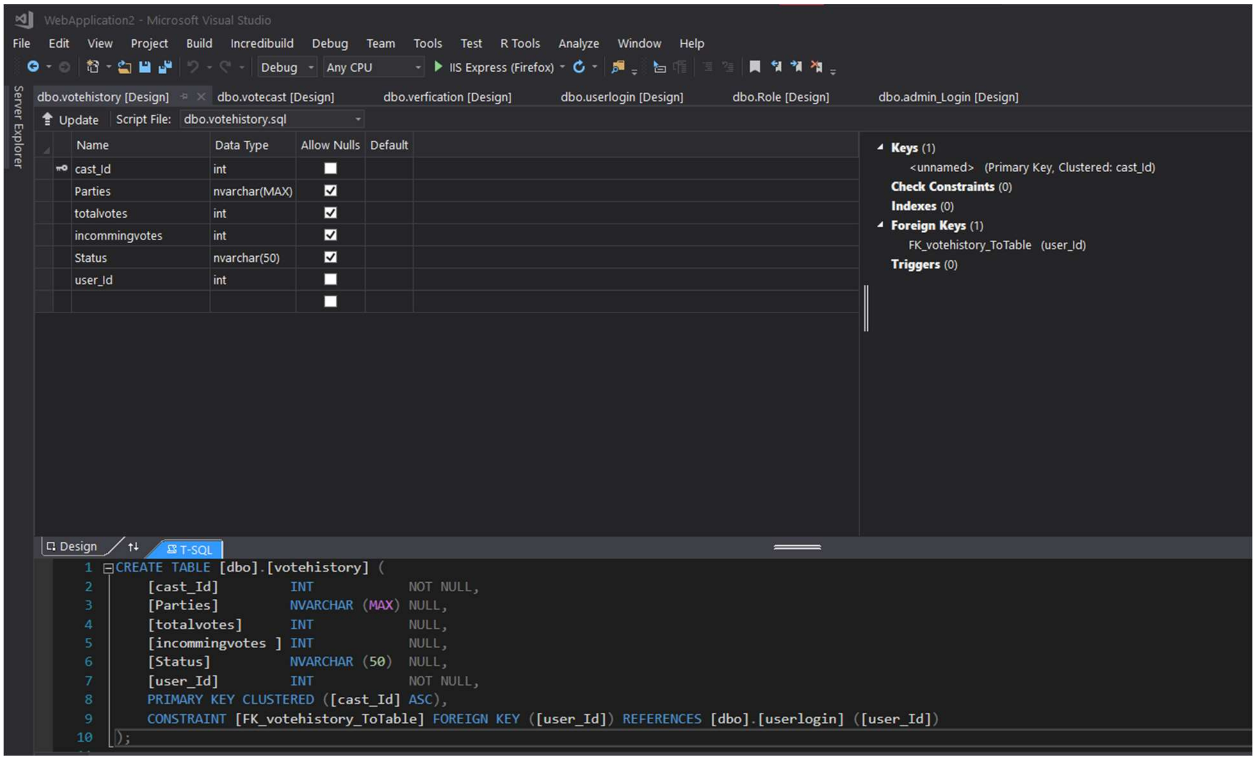
Keys (1)
<unnamed> (Primary Key, Clustered: cast_vote_id)
Check Constraints (0)
Indexes (0)
Foreign Keys (1)
FK_votecast_ToTable (user_Id)
Triggers (0)

```
CREATE TABLE [dbo].[votecast] (
    [cast_vote_id] INT            NOT NULL,
    [user_Id]      INT            NOT NULL,
    [party]        NVARCHAR (50)  NULL,
    [post]         NVARCHAR (50)  NULL,
    [email]        NVARCHAR (MAX) NULL,
    PRIMARY KEY CLUSTERED ([cast_vote_id] ASC),
    CONSTRAINT [FK_votecast_ToTable] FOREIGN KEY ([user_Id]) REFERENCES [dbo].[userlogin] ([user_Id])
);
```

Figure 7: Vote cast Table

Vote History:

```
CREATE TABLE [dbo].[votehistory] (
    [cast_Id]           INT            NOT NULL,
    [Parties]           NVARCHAR (MAX) NULL,
    [totalvotes]        INT            NULL,
    [incommingvotes ] INT            NULL,
    [Status]            NVARCHAR (50)  NULL,
    [user_Id]           INT            NOT NULL,
    PRIMARY KEY CLUSTERED ([cast_Id] ASC),
    CONSTRAINT [FK_votehistory_ToTable] FOREIGN KEY ([user_Id]) REFERENCES
[dbo].[userlogin] ([user_Id])
);
```



Script File: dbo.votehistory.sql

Name	Data Type	Allow Nulls	Default
cast_Id	int	□	
Parties	nvarchar(MAX)	☑	
totalvotes	int	☑	
incommingvotes	int	☑	
Status	nvarchar(50)	☑	
user_Id	int	□	
		□	

- **Keys** (1)
  - <unnamed> (Primary Key, Clustered: cast_Id)
- **Check Constraints** (0)
- **Indexes** (0)
- **Foreign Keys** (1)
  - FK_votehistory_ToTable (user_Id)
- **Triggers** (0)

```sql
CREATE TABLE [dbo].[votehistory] (
    [cast_Id]          INT            NOT NULL,
    [Parties]          NVARCHAR (MAX) NULL,
    [totalvotes]       INT            NULL,
    [incommingvotes ] INT            NULL,
    [Status]           NVARCHAR (50)  NULL,
    [user_Id]          INT            NOT NULL,
    PRIMARY KEY CLUSTERED ([cast_Id] ASC),
    CONSTRAINT [FK_votehistory_ToTable] FOREIGN KEY ([user_Id]) REFERENCES [dbo].[userlogin] ([user_Id])
);
```

Figure 8: Vote History Table

Web Pages:

Index.html:

```html
<!DOCTYPE html>
<html>
    <head>
        <title>PEC</title>
        <link rel="stylesheet" href="Stylesheet.css" type="text/css">
        <script src="../Scripts/bootstrap.js"></script>
        <script src="../Scripts/jquery-3.3.1.js"></script>
    </head>
    <body>
        <div class="banner">
            <div class="navbar">
                <img src="Images/logo.png" class="logo">
                <ul>
                    <li><a href="index.html">Home</a></li>
                    <li><a href="Parties/New/Parties.html">Parties</a></li>
                    <li><a href="Viewer/Viwer.html">Account</a></li>
                </ul>
            </div>
            <div class="content">
                <h1 class="hover-underline-animation">Pakistan Voting 2023</h1>
                <br>
                <p class="hover-underline-animation">Election Commission of Pakistan,
PEC</p>
                <div>
                    <a href="Cast_Vote/cast.html"><button type="button"><span></span>Cast
Vote</button></a>
```

```
                    <a href="Verify/verfiy.html"><button
type="button"><span></span>Verify ID</button></a>
                </div>
            </div>
        </div>
    </body>
</html>
```

User_Login.html:

```
<html>
<head>
    <meta charset="utf-8">
    <title>Login and Registtation Form</title>
    <link rel="stylesheet" href="styles.css">
    <script src="../../../../../Scripts/bootstrap.js"></script>
    <script src="../../../../../Scripts/jquery-3.3.1.js"></script>
    <link href="../../../../../Content/bootstrap.css" rel="stylesheet" />
</head>
<body ng-app="myModule">
    <div class="hero" ng-controller="loginController">
        <div class="form-box">
            <div class="button-box">
                <div id="btn"></div>
                <button type="button" class="toggle-btn" onclick="login()">Log
In</button>
                <button type="button" class="toggle-btn"
onclick="register()">Register</button>
            </div>
            <div class="social-icons">
                <img src="fb.png">
                <img src="tw.png">
                <img src="gp.png">
            </div>
            <form action="" method="post" id="login" class="input-group" ng-
controller="loginController">
                <input type="text" class="input-field" placeholder="User Id"
name="username" ng-model="email" required>


                <input type="text" class="input-field" placeholder="Enter Password"
name="password" ng-model="password" required>
                <br>
                <input type="checkbox" class="check-box"><span>Remember Password</span>
                <a href="../dashboard.html">
                    <button type="button" class="submit-btn" ng-click="loginUser()>Log
in</button>
                </a>
            </form>

            <form id="register" class="input-group">
                <input type="text" class="input-field" placeholder="User Id" ng-
model="username" required>
                <input type="email" class="input-field" placeholder="Email Id" ng-
model="email" required>
                <input type="text" class="input-field" placeholder="Enter Password" ng-
model="password" required>
```

```
                <br>
                <input type="checkbox" class="check-box"><span>I agree to the Terms and
Conditions </span>
                <button type="submit" class="submit-btn" ng-
click="register()">Register</button>
            </form>
        </div>
    </div>


    <script>
        var x = document.getElementById("login");
        var y = document.getElementById("register");
        var z = document.getElementById("btn");

        function register() {
            x.style.left = "-400px";
            y.style.left = "50px";
            z.style.left = "110px";
        }

        function login() {
            x.style.left = "50px";
            y.style.left = "450px";
            z.style.left = "0px";
        }
    </script>

    <script>
        var myApp = angular.module('myApp', []);
        myApp.controller("regUserController", function ($scope, $http) {
            $scope.register = function () {
                var bearer = "";
                var user = {
                    grant_type: 'password',
                    Email: $scope.email,
                    password: $scope.password,
                    UserId: $scope.UserId
                };
                $.ajax({
                    type: "POST",
                    url: "https://localhost:44339/api/Account/Register",
                    data: user,
                    contentType: "application/x-www-form-urlencoded",
                    dataType: "text",
                    success: function (data) {
                        alert("registered successfully");
                    },
                    failure: function (response) {
                        alert("failed " + JSON.stringify(response));
                    },
                    error: function (response) {
                        alert("On Error " + JSON.stringify(response));
                    }
                });
            }
        });
        myApp.controller("loginController", function ($scope, $http) {
            $scope.signIn = function () {
                var bearer = "";
                var user = {
                    grant_type: 'password',
                    username: $scope.email,
                    password: $scope.password
                };
                $.ajax({
                    type: "POST",
                    url: "https://localhost:44339/token",
                    data: user,
                    contentType: "application/x-www-form-urlencoded",
                    dataType: "json",
                    success: function (data) {
                        bearer = data.access_token;
                        localStorage.setItem('token', bearer);
                        window.location.href =<a
href="../dashboard.html">../dashboard.html</a>;
                    },
                    failure: function (response) {
                        alert("Failed" + response.responseText);
                        alert(JSON.stringify(response));
                    },
                    error: function (response) {
                        alert("Error" + response.responseText);
                        alert(JSON.stringify(response));
                    }
                });
            }
        });
    </script>

</body>
</html>
```

User Dashboard:

```
<!DOCTYPE html>

<html>



<head>

 <meta charset="utf-8">

 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

 <meta name="description" content="Start your development with a Dashboard for Bootstrap 4.">

 <meta name="author" content="Creative Tim">

 <title>User Dashboard</title>

 <!-- Favicon -->
 <link rel="icon" href="../assets/img/brand/favicon.png" type="image/png">
 <!-- Fonts -->
 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700">
 <!-- Icons -->
 <link rel="stylesheet" href="../assets/vendor/nucleo/css/nucleo.css" type="text/css">
 <link rel="stylesheet" href="../assets/vendor/@fortawesome/fontawesome-free/css/all.min.css"
type="text/css">
 <!-- Page plugins -->
 <!-- Argon CSS -->
 <link rel="stylesheet" href="../assets/css/argon.css?v=1.2.0" type="text/css">
 <script src="../../../../Scripts/bootstrap.js"></script>
 <script src="../../../../Scripts/jquery-3.3.1.js"></script>
 <link href="../../../../Content/bootstrap.css" rel="stylesheet" />
</head>


<body>
 <!-- Sidenav -->
 <nav class="sidenav navbar navbar-vertical  fixed-left  navbar-expand-xs navbar-light bg-white"
id="sidenav-main">
   <div class="scrollbar-inner">
    <!-- Brand -->
    <div class="sidenav-header  align-items-center">
     <a class="navbar-brand" href="javascript:void(0)">
      <img src="../assets/img/brand/blue.png" class="navbar-brand-img" alt="...">
     </a>
    </div>
    <div class="navbar-inner">
     <!-- Collapse -->
     <div class="collapse navbar-collapse" id="sidenav-collapse-main">
```

```
        <!-- Nav items -->
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link active" href="dashboard.html">
              <i class="ni ni-tv-2 text-primary"></i>
              <span class="nav-link-text">Dashboard</span>
            </a>
          </li>


          <li class="nav-item">
            <a class="nav-link" href="profile.html">
              <i class="ni ni-single-02 text-yellow"></i>
              <span class="nav-link-text">Profile</span>
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="tables.html">
              <i class="ni ni-bullet-list-67 text-default"></i>
              <span class="nav-link-text">Tables</span>
            </a>
          </li>
        </ul>
      </div>
    </div>
  </div>
</nav>
<!-- Main content -->
<div class="main-content" id="panel">
```

```
  <!-- Topnav -->
  <nav class="navbar navbar-top navbar-expand navbar-dark bg-primary border-bottom">
   <div class="container-fluid">
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
     <!-- Search form -->
     <form class="navbar-search navbar-search-light form-inline mr-sm-3" id="navbar-search-main">
      <div class="form-group mb-0">
       <div class="input-group input-group-alternative input-group-merge">
        <div class="input-group-prepend">
         <span class="input-group-text"><i class="fas fa-search"></i></span>
        </div>
        <input class="form-control" placeholder="Search" type="text">
       </div>
      </div>
      <button type="button" class="close" data-action="search-close" data-target="#navbar-search-main" aria-label="Close">
       <span aria-hidden="true">×</span>
      </button>
     </form>
     <!-- Navbar links -->
     <ul class="navbar-nav align-items-center  ml-md-auto ">
      <li class="nav-item d-xl-none">
       <!-- Sidenav toggler -->
       <div class="pr-3 sidenav-toggler sidenav-toggler-dark" data-action="sidenav-pin" data-target="#sidenav-main">
        <div class="sidenav-toggler-inner">
         <i class="sidenav-toggler-line"></i>
         <i class="sidenav-toggler-line"></i>
         <i class="sidenav-toggler-line"></i>
    </div>
  </div>
</li>
<li class="nav-item d-sm-none">
  <a class="nav-link" href="#" data-action="search-show" data-target="#navbar-search-main">
    <i class="ni ni-zoom-split-in"></i>
  </a>
</li>
<li class="nav-item dropdown">
  <a class="nav-link" href="#" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    <i class="ni ni-bell-55"></i>
  </a>
  <div class="dropdown-menu dropdown-menu-xl  dropdown-menu-right  py-0 overflow-hidden">
    <!-- Dropdown header -->
    <div class="px-3 py-3">
      <h6 class="text-sm text-muted m-0">You have <strong class="text-primary">13</strong> notifications.</h6>
    </div>
    <!-- List group -->
    <div class="list-group list-group-flush">
      <a href="#!" class="list-group-item list-group-item-action">
        <div class="row align-items-center">
          <div class="col-auto">
            <!-- Avatar -->
            <img alt="Image placeholder" src="../assets/img/theme/team-1.jpg" class="avatar rounded-circle">
          </div>
          <div class="col ml--2">
            <div class="d-flex justify-content-between align-items-center">
```

```
             <div>
              <h4 class="mb-0 text-sm">GOP</h4>
             </div>
             <div class="text-right text-muted">
              <small></small>
             </div>
            </div>
            <p class="text-sm mb-0">Votting Start at 13th April</p>
           </div>
          </div>
         </a>
         <a href="#!" class="list-group-item list-group-item-action">
          <div class="row align-items-center">
           <div class="col-auto">
            <!-- Avatar -->
            <img alt="Image placeholder" src="../assets/img/theme/team-2.jpg" class="avatar
rounded-circle">
           </div>
           <div class="col ml--2">
            <div class="d-flex justify-content-between align-items-center">
             <div>
              <h4 class="mb-0 text-sm">PEC</h4>
             </div>
             <div class="text-right text-muted">
              <small>3 hrs ago</small>
             </div>
            </div>
            <p class="text-sm mb-0">The results are here.</p>
           </div>
```

```
 </div>
</a>
<a href="#!" class="list-group-item list-group-item-action">
 <div class="row align-items-center">
  <div class="col-auto">
   <!-- Avatar -->
   <img alt="Image placeholder" src="../assets/img/theme/team-3.jpg" class="avatar rounded-circle">
  </div>
  <div class="col ml--2">
   <div class="d-flex justify-content-between align-items-center">
    <div>
     <h4 class="mb-0 text-sm">PEC</h4>
    </div>
    <div class="text-right text-muted">
     <small>5 hrs ago</small>
    </div>
   </div>
   <p class="text-sm mb-0">Your posts have been liked a lot.</p>
  </div>
 </div>
</a>
<a href="#!" class="list-group-item list-group-item-action">
 <div class="row align-items-center">
  <div class="col-auto">
   <!-- Avatar -->
   <img alt="Image placeholder" src="../assets/img/theme/team-4.jpg" class="avatar rounded-circle">
  </div>
```

```
          <div class="col ml--2">
            <div class="d-flex justify-content-between align-items-center">
              <div>
                <h4 class="mb-0 text-sm">GOP</h4>
              </div>
              <div class="text-right text-muted">
                <small>2 hrs ago</small>
              </div>
            </div>
            <p class="text-sm mb-0">There was a problem so election will be held again.</p>
          </div>
        </div>
      </a>
      <a href="#!" class="list-group-item list-group-item-action">
        <div class="row align-items-center">
          <div class="col-auto">
            <!-- Avatar -->
            <img alt="Image placeholder" src="../assets/img/theme/team-5.jpg" class="avatar
rounded-circle">
          </div>
          <div class="col ml--2">
            <div class="d-flex justify-content-between align-items-center">
              <div>
                <h4 class="mb-0 text-sm">GOP</h4>
              </div>
              <div class="text-right text-muted">
                <small>3 hrs ago</small>
              </div>
            </div>
            <p class="text-sm mb-0">Service will be down for an hour.</p>
           </div>
          </div>
         </a>
        </div>
        <!-- View all -->
        <a href="#!" class="dropdown-item text-center text-primary font-weight-bold py-3">View
all</a>
       </div>
      </li>

     </ul>
     <ul class="navbar-nav align-items-center  ml-auto ml-md-0 ">
      <li class="nav-item dropdown">
       <a class="nav-link pr-0" href="#" role="button" data-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">
        <div class="media align-items-center">
         <span class="avatar avatar-sm rounded-circle">
          <img alt="Image placeholder" src="../assets/img/theme/team-4.jpg">
         </span>
         <div class="media-body  ml-2  d-none d-lg-block">
          <span class="mb-0 text-sm  font-weight-bold">User</span>
         </div>
        </div>
       </a>
       <div class="dropdown-menu  dropdown-menu-right ">
        <div class="dropdown-header noti-title">
         <h6 class="text-overflow m-0">Welcome!</h6>
        </div>
        <a href="profile.html" class="dropdown-item">
         <i class="ni ni-single-02"></i>
         <span>My profile</span>
        </a>
        <a href="#!" class="dropdown-item">
         <i class="ni ni-settings-gear-65"></i>
         <span>Settings</span>
        </a>
        <a href="#!" class="dropdown-item">
         <i class="ni ni-calendar-grid-58"></i>
         <span>Activity</span>
        </a>
        <div class="dropdown-divider"></div>
        <a href="userlogin/Login.html" class="dropdown-item">
         <i class="ni ni-user-run"></i>
         <span>Logout</span>
        </a>
       </div>
      </li>
     </ul>
    </div>
   </div>
  </nav>
  <!-- Header -->
  <!-- Header -->
  <div class="header bg-primary pb-6">
   <div class="container-fluid">
    <div class="header-body">
     <div class="row align-items-center py-4">
    <div class="col-lg-6 col-7">
      <h6 class="h2 text-white d-inline-block mb-0">Default</h6>
      <nav aria-label="breadcrumb" class="d-none d-md-inline-block ml-md-4">
        <ol class="breadcrumb breadcrumb-links breadcrumb-dark">
          <li class="breadcrumb-item"><a href="#"><i class="fas fa-home"></i></a></li>
          <li class="breadcrumb-item"><a href="#">Dashboards</a></li>
          <li class="breadcrumb-item active" aria-current="page">Default</li>
        </ol>
      </nav>
    </div>
    <div class="col-lg-6 col-5 text-right">
      <a href="#" class="btn btn-sm btn-neutral">New</a>
      <a href="#" class="btn btn-sm btn-neutral">Filters</a>
    </div>
  </div>
  <!-- Card stats -->
  <div class="row">
    <div class="col-xl-3 col-md-6">
      <div class="card card-stats">
        <!-- Card body -->
        <div class="card-body">
          <div class="row">
            <div class="col">
              <h5 class="card-title text-uppercase text-muted mb-0">Total Votes</h5>
              <span class="h2 font-weight-bold mb-0">350,897,0000</span>
            </div>
            <div class="col-auto">
              <div class="icon icon-shape bg-gradient-red text-white rounded-circle shadow">
                <i class="ni ni-active-40"></i>
              </div>
            </div>
          </div>
          <p class="mt-3 mb-0 text-sm">
            <span class="text-success mr-2"><i class="fa fa-arrow-up"></i> 15.48%</span>
            <span class="text-nowrap">Since last month</span>
          </p>
        </div>
      </div>
    </div>
    <div class="col-xl-3 col-md-6">
      <div class="card card-stats">
        <!-- Card body -->
        <div class="card-body">
          <div class="row">
            <div class="col">
              <h5 class="card-title text-uppercase text-muted mb-0">New User Votes</h5>
              <span class="h2 font-weight-bold mb-0">2,356</span>
            </div>
            <div class="col-auto">
              <div class="icon icon-shape bg-gradient-orange text-white rounded-circle shadow">
                <i class="ni ni-chart-pie-35"></i>
              </div>
            </div>
          </div>
          <p class="mt-3 mb-0 text-sm">
            <span class="text-success mr-2"><i class="fa fa-arrow-up"></i> 3.48%</span>
            <span class="text-nowrap">Since last month</span>
          </p>
      </div>
     </div>
    </div>
    <div class="col-xl-3 col-md-6">
     <div class="card card-stats">
      <!-- Card body -->
      <div class="card-body">
       <div class="row">
        <div class="col">
         <h5 class="card-title text-uppercase text-muted mb-0">Total Number of Votes in.</h5>
         <span class="h2 font-weight-bold mb-0">924</span>
        </div>
        <div class="col-auto">
         <div class="icon icon-shape bg-gradient-green text-white rounded-circle shadow">
          <i class="ni ni-money-coins"></i>
         </div>
        </div>
       </div>
       <p class="mt-3 mb-0 text-sm">
        <span class="text-success mr-2"><i class="fa fa-arrow-up"></i> 3.48%</span>
        <span class="text-nowrap">Since last day</span>
       </p>
      </div>
     </div>
    </div>
    <div class="col-xl-3 col-md-6">
     <div class="card card-stats">
      <!-- Card body -->
      <div class="card-body">
```

```
       <div class="row">
        <div class="col">
         <h5 class="card-title text-uppercase text-muted mb-0">Performance of Parties</h5>
         <span class="h2 font-weight-bold mb-0">49,65%</span>
        </div>
        <div class="col-auto">
         <div class="icon icon-shape bg-gradient-info text-white rounded-circle shadow">
          <i class="ni ni-chart-bar-32"></i>
         </div>
        </div>
       </div>
       <p class="mt-3 mb-0 text-sm">
        <span class="text-success mr-2"><i class="fa fa-arrow-up"></i> 3.48%</span>
        <span class="text-nowrap">Since last month</span>
       </p>
      </div>
     </div>
    </div>
   </div>
  </div>
 </div>
</div>
<!-- Page content -->
<div class="container-fluid mt--6">
 <div class="row">
  <div class="col-xl-8">
   <div class="card bg-default">
    <div class="card-header bg-transparent">
     <div class="row align-items-center">
```

```
        <div class="col">
          <h6 class="text-light text-uppercase ls-1 mb-1">Overview</h6>
          <h5 class="h3 text-white mb-0">Votes</h5>
        </div>
        <div class="col">
          <ul class="nav nav-pills justify-content-end">
            <li class="nav-item mr-2 mr-md-0" data-toggle="chart" data-target="#chart-sales-dark" data-update='{"data":{"datasets":[{"data":[0, 20, 10, 30, 15, 40, 20, 60, 60]}]}}' data-prefix="" data-suffix="k">
              <a href="#" class="nav-link py-2 px-3 active" data-toggle="tab">
                <span class="d-none d-md-block">Month</span>
                <span class="d-md-none">M</span>
              </a>
            </li>
            <li class="nav-item" data-toggle="chart" data-target="#chart-sales-dark" data-update='{"data":{"datasets":[{"data":[0, 20, 5, 25, 10, 30, 15, 40, 40]}]}}' data-prefix="" data-suffix="k">
              <a href="#" class="nav-link py-2 px-3" data-toggle="tab">
                <span class="d-none d-md-block">Week</span>
                <span class="d-md-none">W</span>
              </a>
            </li>
          </ul>
        </div>
      </div>
    </div>
    <div class="card-body">
      <!-- Chart -->
      <div class="chart">
        <!-- Chart wrapper -->
        <canvas id="chart-sales-dark" class="chart-canvas"></canvas>
        </div>
      </div>
    </div>
  </div>
  <div class="col-xl-4">
    <div class="card">
      <div class="card-header bg-transparent">
        <div class="row align-items-center">
          <div class="col">
            <h6 class="text-uppercase text-muted ls-1 mb-1">Performance</h6>
            <h5 class="h3 mb-0">Total Votes</h5>
          </div>
        </div>
      </div>
      <div class="card-body">
        <!-- Chart -->
        <div class="chart">
          <canvas id="chart-bars" class="chart-canvas"></canvas>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-xl-8">
    <div class="card">
      <div class="card-header border-0">
        <div class="row align-items-center">
          <div class="col">
```

```
   <h3 class="mb-0">New Sites traffic</h3>
  </div>
 </div>
</div>
<div class="table-responsive">
 <!-- Projects table -->
 <table class="table align-items-center table-flush">
  <thead class="thead-light">
   <tr>
    <th scope="col">Page name</th>
    <th scope="col">Visitors</th>
    <th scope="col">Unique users</th>
    <th scope="col">Bounce rate</th>
   </tr>
  </thead>
  <tbody>
   <tr>
    <th scope="row">
     Ary News <br>
     <h6>https://arynews.tv</h6>
    </th>
    <td>
     4,569
    </td>
    <td>
     340
    </td>
    <td>
     <i class="fas fa-arrow-up text-success mr-3"></i> 46,53%
    </td>
   </tr>
   <tr>
    <th scope="row">
    Geo News
    <br>
    <h6>https://www.geo.tv/</h6>
    </th>
    <td>
     3,985
    </td>
    <td>
     319
    </td>
    <td>
     <i class="fas fa-arrow-down text-warning mr-3"></i> 46,53%
    </td>
   </tr>
   <tr>
    <th scope="row">
     24 News
    <br>
    <h6>https://www.geo.tv/</h6>
    </th>
    <td>
     3,513
    </td>
    <td>
     294
    </td>
    <td>
     <i class="fas fa-arrow-down text-warning mr-3"></i> 36,49%
    </td>
   </tr>
   <tr>
    <th scope="row">
     92 News
      <br>
    <h6>https://www.92news.tv/</h6>
    </th>
    <td>
     2,050
    </td>
    <td>
     147
    </td>
    <td>
     <i class="fas fa-arrow-up text-success mr-3"></i> 50,87%
    </td>
   </tr>
   <tr>
    <th scope="row">
     Bol News
      <br>
    <h6>https://www.bol.tv/</h6>
    </th>
    <td>
     1,795
```

```
            </td>
            <td>
              190
            </td>
            <td>
              <i class="fas fa-arrow-down text-danger mr-3"></i> 46,53%
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>
<div class="col-xl-4">
  <div class="card">
    <div class="card-header border-0">
      <div class="row align-items-center">
        <div class="col">
          <h3 class="mb-0">Social traffic</h3>
        </div>
      </div>
    </div>
    <div class="table-responsive">
      <!-- Projects table -->
      <table class="table align-items-center table-flush">
        <thead class="thead-light">
          <tr>
            <th scope="col">Referral</th>
            <th scope="col">Visitors</th>
            <th scope="col"></th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <th scope="row">
              Facebook
            </th>
            <td>
              1,480
            </td>
            <td>
              <div class="d-flex align-items-center">
                <span class="mr-2">60%</span>
                <div>
                  <div class="progress">
                    <div class="progress-bar bg-gradient-danger" role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-valuemax="100" style="width: 60%;"></div>
                  </div>
                </div>
              </div>
            </td>
          </tr>
          <tr>
            <th scope="row">
              Facebook
            </th>
            <td>
              5,480
            </td>
            <td>
              <div class="d-flex align-items-center">
                <span class="mr-2">70%</span>
                <div>
                  <div class="progress">
                    <div class="progress-bar bg-gradient-success" role="progressbar" aria-valuenow="70" aria-valuemin="0" aria-valuemax="100" style="width: 70%;"></div>
                  </div>
                </div>
              </div>
            </td>
          </tr>
          <tr>
            <th scope="row">
              Google
            </th>
            <td>
              4,807
            </td>
            <td>
              <div class="d-flex align-items-center">
                <span class="mr-2">80%</span>
                <div>
                  <div class="progress">
                    <div class="progress-bar bg-gradient-primary" role="progressbar" aria-valuenow="80" aria-valuemin="0" aria-valuemax="100" style="width: 80%;"></div>
                  </div>
                </div>
              </div>
            </td>
          </tr>
          <tr>
            <th scope="row">
              Instagram
            </th>
            <td>
              3,678
            </td>
            <td>
              <div class="d-flex align-items-center">
                <span class="mr-2">75%</span>
                <div>
                  <div class="progress">
                    <div class="progress-bar bg-gradient-info" role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100" style="width: 75%;"></div>
                  </div>
                </div>
              </div>
            </td>
          </tr>
          <tr>
            <th scope="row">
              twitter
            </th>
            <td>
              2,645
            </td>
```

```
          <td>
           <div class="d-flex align-items-center">
            <span class="mr-2">30%</span>
            <div>
             <div class="progress">
              <div class="progress-bar bg-gradient-warning" role="progressbar" aria-valuenow="30" aria-valuemin="0" aria-valuemax="100" style="width: 30%;"></div>
             </div>
            </div>
           </div>
          </td>
         </tr>
        </tbody>
       </table>
      </div>
     </div>
    </div>
   </div>
   <!-- Footer -->
   <footer class="footer pt-0">
    <div class="row align-items-center justify-content-lg-between">
     <div class="col-lg-6">
      <div class="copyright text-center  text-lg-left  text-muted">
       &copy; 2020 <a href="" class="font-weight-bold ml-1" target="_blank">Abdul Rafay & Haseeb Ahmad</a>
      </div>
     </div>
     <div class="col-lg-6">
      <ul class="nav nav-footer justify-content-center justify-content-lg-end">
```

```
              <li class="nav-item">
                <a href="" class="nav-link" target="_blank">Abdul Rafay & Haseeb Ahmed</a>
              </li>
              <li class="nav-item">
                <a href="" class="nav-link" target="_blank">About Us</a>
              </li>
              <li class="nav-item">
                <a href="ContactFrom_v15/index.html" class="nav-link" target="_blank">Contact Us</a>
              </li>

            </ul>
          </div>
        </div>
      </footer>
    </div>
  </div>
  <!-- Argon Scripts -->
  <!-- Core -->
  <script src="../assets/vendor/jquery/dist/jquery.min.js"></script>
  <script src="../assets/vendor/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
  <script src="../assets/vendor/js-cookie/js.cookie.js"></script>
  <script src="../assets/vendor/jquery.scrollbar/jquery.scrollbar.min.js"></script>
  <script src="../assets/vendor/jquery-scroll-lock/dist/jquery-scrollLock.min.js"></script>
  <!-- Optional JS -->
  <script src="../assets/vendor/chart.js/dist/Chart.min.js"></script>
  <script src="../assets/vendor/chart.js/dist/Chart.extension.js"></script>
  <!-- Argon JS -->
  <script src="../assets/js/argon.js?v=1.2.0"></script>
</body>
```

</html>

Admin Login:

```
<!doctype html>
<html lang="en">
<head>
    <title>Admin Login</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">

    <script src="../../../Scripts/bootstrap.js"></script>
    <script src="../../../Scripts/bootstrap.min.js"></script>
    <script src="../../../Scripts/jquery-3.3.1.js"></script>
    <script src="../../../Scripts/jquery.min.js"></script>

    <link rel="stylesheet" href="css/style.css" type="text/css">
</head>
       <body class="img js-fullheight" style="background-image: url(images/bg.jpg);">
       <section class="ftco-section">
              <div class="container">
                     <div class="row justify-content-center">
                            <div class="col-md-6 text-center mb-5">
                                   <h2 class="heading-section">Admin Login</h2>
                            </div>
                     </div>
                     <div class="row justify-content-center">
                            <div class="col-md-6 col-lg-4">
                                   <div class="login-wrap p-0">
                     <h3 class="mb-4 text-center">Have an account?</h3>
                     <form action="#" class="signin-form">
                            <div class="form-group">
                                   <input type="text" class="form-control"
placeholder="Username" required>
                            </div>
                   <div class="form-group">
                     <input id="password-field" type="password" class="form-control"
placeholder="Password" required>
                     <span toggle="#password-field" class="fa fa-fw fa-eye field-icon
toggle-password"></span>
                   </div>

                   <div class="form-group">
                     <a href="dashboard/AdminDashBoard.html">
                        <button type="button" class="form-control btn btn-primary submit
px-3">Sign In</button>
                      </a>

                    </div>
```

```
                 </form>


                    </div>
                            </div>
                     </div>
              </div>
       </section>

  <script src="js/jquery.min.js"></script>
  <script src="js/popper.js"></script>
  <script src="js/bootstrap.min.js"></script>
  <script src="js/main.js"></script>

       </body>
</html>
```

Admin Dashboard:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css"
      integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="stylesAD.css" type="text/css" />
    <script src="../../../../Scripts/bootstrap.min.js"></script>
    <script src="../../../../Scripts/jquery-3.3.1.js"></script>
    <link href="../../../../Content/bootstrap.css" rel="stylesheet" />
    <title>Admin Dashboard</title>
  </head>
  <body id="body">
    <div class="container">
      <nav class="navbar">
        <div class="nav_icon" onclick="toggleSidebar()">
          <i class="fa fa-bars" aria-hidden="true"></i>
        </div>
        <div class="navbar__left">
          <a class="active_link" href="#">Admin</a>
        </div>
        <div class="navbar__right">
          <a href="#">
            <i class="fa fa-search" aria-hidden="true"></i>
          </a>
          <a href="#">
            <i class="fa fa-clock-o" aria-hidden="true"></i>
          </a>
          <a href="#">
            <img width="30" src="assets/logo.png" alt="" />
            <!-- <i class="fa fa-user-circle-o" aria-hidden="true"></i> -->
          </a>
        </div>
```

```
</nav>

<main>
  <div class="main__container">
    <!-- MAIN TITLE STARTS HERE -->

    <div class="main__title">
      <img src="assets/logo.png" alt="" />
      <div class="main__greeting">
        <h1>Hello Admin</h1>
        <p>Welcome to your admin dashboard</p>
      </div>
    </div>

    <!-- MAIN TITLE ENDS HERE -->

    <!-- MAIN CARDS STARTS HERE -->
    <div class="main__cards">
      <div class="card">
        <i
          class="fa fa-user-o fa-2x text-lightblue"
          aria-hidden="true"
        ></i>
        <div class="card_inner">
          <p class="text-primary-p">Total votes casted</p>
          <span class="font-bold text-title">578</span>
        </div>
      </div>

      <div class="card">
        <i class="fa fa-calendar fa-2x text-red" aria-hidden="true"></i>
        <div class="card_inner">
          <p class="text-primary-p">Yearly Voting Schedule</p>
          <span class="font-bold text-title">2467</span>
        </div>
      </div>

      <div class="card">
        <i
          class="fa fa-video-camera fa-2x text-yellow"
          aria-hidden="true"
        ></i>
        <div class="card_inner">
          <p class="text-primary-p">Live coverage</p>
          <span class="font-bold text-title">340</span>
        </div>
      </div>

      <div class="card">
        <i
          class="fa fa-thumbs-up fa-2x text-green"
          aria-hidden="true"
        ></i>
        <div class="card_inner">
          <p class="text-primary-p">Postive voter review</p>
          <span class="font-bold text-title">645</span>
        </div>
      </div>
```

```
        </div>
        <!-- MAIN CARDS ENDS HERE -->

        <!-- CHARTS STARTS HERE -->
        <div class="charts">
          <div class="charts__left">
            <div class="charts__left__title">
              <div>
                <h1>Daily Reports</h1>
                <p>Pakistan</p>
              </div>

            </div>
            <div id="apex1"></div>
          </div>

          <div class="charts__right">
            <div class="charts__right__title">
              <div>
                <h1>Election Status</h1>
                <p>Pakistan</p>
              </div>

            </div>

            <div class="charts__right__cards">
              <div class="card1">
                <h1>2018</h1>
                <p>115,748,753</p>
              </div>

              <div class="card2">
                <h1>2009</h1>
                <p>14,794,188</p>
              </div>

              <div class="card3">
                <h1>2004</h1>
                <p>16903702</p>
              </div>

              <div class="card4">
                <h1>2000</h1>
                <p>14587993</p>
              </div>
            </div>
          </div>
        </div>
        <!-- CHARTS ENDS HERE -->
      </div>
    </main>

    <div id="sidebar">
      <div class="sidebar__title">
        <div class="sidebar__img">
          <img src="assets/logo.png" alt="logo" />
          <h1>&nbsp PEC </h1>
        </div>
        <i
          onclick="closeSidebar()"
          class="fa fa-times"
          id="sidebarIcon"
          aria-hidden="true"
        ></i>
      </div>

      <div class="sidebar__menu">
        <div class="sidebar__link active_menu_link">
          <i class="fa fa-home"></i>
          <a href="#">Dashboard</a>
        </div>
        <h2>MNG</h2>
        <div class="sidebar__link">
          <i class="fa fa-user-secret" aria-hidden="true"></i>
          <a href="#">Admin Management</a>
        </div>
        <div class="sidebar__link">
          <i class="fa fa-building-o"></i>
          <a href="#">Provinces</a>
        </div>
        <div class="sidebar__link">
          <i class="fa fa-wrench"></i>
          <a href="#">Casted Vote</a>
        </div>
        <div class="sidebar__link">
          <i class="fa fa-archive"></i>
          <a href="#">Start Voting</a>
        </div>
        <div class="sidebar__link">
          <i class="fa fa-handshake-o"></i>
          <a href="#">End Voting</a>
        </div>
        <h2>LEAVE</h2>
        <div class="sidebar__link">
          <i class="fa fa-question"></i>
          <a href="#">Requests</a>
        </div>
        <div class="sidebar__link">
          <i class="fa fa-sign-out"></i>
          <a href="#">Issues</a>
        </div>
        <h2>Maintanance</h2>
        <div class="sidebar__link">
          <i class="fa fa-money"></i>
          <a href="#">Maintanance</a>
        </div>
        <div class="sidebar__link">
          <i class="fa fa-briefcase"></i>
          <a href="#">Contacts</a>
        </div>
        <div class="sidebar__logout">
          <i class="fa fa-power-off"></i>
          <a href="AdminLogin.html">Log out</a>
        </div>
      </div>
    </div>
```

```
    </div>
    <script src="https://cdn.jsdelivr.net/npm/apexcharts"></script>
    <script src="script.js"></script>
  </body>
</html>
```

Viewer Page:

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Dashboard</title>
    <script src="../../Scripts/bootstrap.js"></script>
    <script src="../../Scripts/jquery-3.3.1.js"></script>
    <link href="../../Content/bootstrap.css" rel="stylesheet" />
    <style>

        html, body {
            font: 400 0.875rem/1.5 "Open Sans", sans-serif;
            margin: 0;
            padding: 0;
            background-color: #eee;
            background-image: url(images/16680.jpg);
            background-repeat: no-repeat;
            background-image: cover;
            background-position: center;
        }

        h1 {
            font-size: 18px;
            text-transform: uppercase;
            color: #eee;
        }

        #container {
            margin: 0 auto;
            width: auto;
            padding: 80px 0 0 0;
            text-align: center;
            border: 0px solid #F00;
        }

        .box {
            background-color: black;
            height: 280px;
            vertical-align: top;
            border-radius: 5px;
            width: 250px;
            display: inline-block;
            margin-left: 10px;
            box-sizing: border-box;
            -moz-box-sizing: border-box;
        }

            .box:hover {
                background-color: coral;
            }

        .grey-bg {
            background-color: #eee;
            padding: 20px 0 20px 0;
            width: 100%;
            position: relative;
            text-align: center;
        }

        .logo-box {
            display: inline-block;
            margin: 0 auto;
            border-radius: 10px;
            width: 150px; /*padding:10px;*/
            margin-top: 10px;
        }

        .arrow-down {
            width: 0px;
            height: 0px;
            border-left: 13px solid transparent;
            border-right: 13px solid transparent;
            border-top: 15px solid #eee;
            margin: 0 auto;
        }

        .title {
            margin: 0;
            vertical-align: top;
            text-align: center;
            font-weight: bold;
            font-size: 42px;
            color: #2a6ecf;
            text-transform: none;
            display: inline-block;
        }

        .homebtn:hover {
            background-color: coral;
        }

        img {
            border-top-left-radius: 5px;
            border-top-right-radius: 5px;
        }

        a {
            text-decoration: none;
            outline: none;
        }
        }
    </style>
</head>

<body>
```

```
   <div class="grey-bg">
              <div style="vertical-align: top; display: inline-block;">
          <img height="110px" width="110px" src="images/MicrosoftTeams-
image.png"></div><br/>
        <div class="title"<span style="color:black;font-size:32px; ">PEC</span></div>
</div>
   <div class="arrow-down"></div>
   <div id="container">
        <a href="AdminLogin/AdminLogin.html">
        <div class="box">
        <div style="border:0px solid #F00; height:200px; position: fixed; left:750px;
top:350px;"> <img src="images/admin_Login.png" style="width:150px;height:200px;" ></div>
        <h1>Admin</h1>
    </div>
    </a>
          <a href="user_Login/examples/userlogin/Login.html">
    <div class="box">
        <h1>User</h1>
        <div style="border:0px solid #F00; height:200px;"> <img src="images/4860876.jpg"
style="width:150px;height:200px;"></div>

    </div>
       </a>
<br><br><br>
<button class="homebtn"><a href="../index.html"><span>Go Back Home</span></a></button>

</div>

<style>
.homebtn {
    background: black;
    border: 0;
    font-size: 14px;
    color: #fff !important;
    font-weight: 700;
    padding: 10px 15px;
    margin: 0 auto;
}
.homebtn a {
    color: #fff;
}
</style>
</body>
</html>
```

Verification Page:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Verfiy Account</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link href="contactform.css" rel="stylesheet">
    <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet"
```

```
          integrity="sha384-
wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN"
crossorigin="anonymous">
    <script src="../../Scripts/bootstrap.js"></script>
    <script src="../../Scripts/jquery-3.3.1.js"></script>
    <link href="../../Content/bootstrap.css" rel="stylesheet" />
</head>
<style>
body{
        background-image: url("Image/background1.jpg");
        background-repeat: no-repeat;
        background-position: center;
        background-size: cover;
        height: 100vh;
        width: 100vh;
}
h2{
        font-family: Helvetica ;
}
</style>
<body>

    <div class="wrapper">
        <h2> Verification</h2>
<form action="#" method="POST">
            <div class="input-group">
                <div class="input-box">
                    <label>Name</label>
                    <input type="text" placeholder="Name" required class="name">
                    <i class="fa fa-user"></i>
                </div>
<div class="input-box">
                    <label>CNIC</label>
                    <input type="text" placeholder="CNIC" required class="name">
                    <i class="fa fa-user"></i>
                </div>
</div>
<div class="input-group">
                <div class="input-box">
                    <label>Email</label>
                    <input type="email" placeholder="Email" required class="name">
                    <i class="fa fa-user"></i>
                </div>
<div class="input-box">
                    <label>Phone Number</label>
                    <input type="tel" placeholder="Phone Number" required class="name">
                    <i class="fa fa-envelope"></i>
                </div>
</div>

<div class="button-group">
                <div class="button-box">
                    <button type="submit"><i class="fa fa-paper-plane"></i> Verify
ID</button>
                </div>
</div>
```

```
</form>
</div>
</body>
</html>
```

Parties Page:

```
<html>
<head>
</head>
<title>Political Parties</title>
<link rel="stylesheet" href="styles.css" type="text/css">
<script src="../../../Scripts/bootstrap.js"></script>
<script src="../../../Scripts/jquery-3.3.1.js"></script>
<link href="../../../Content/bootstrap.css" rel="stylesheet" />
<body>
    <div class="banner">
        <div class="Heading"></div>
        <div class="header">
            <h1 class="hover-underline-animation">Political Parties</h1>

        </div>
        <table>
            <tr>
                <th>Party Symbol </th>
                <th>Party Name</th>
                <th>Party History</th>
            </tr>

            <tr>
                <td><img src="Images/1200px-Pakistan_Tehreek-e-Insaf_logo.svg.png"
class="Image"></td>
                <td>Pakistan Tehreek-e-Insaf   انصاف تحریکِ اکستان</td>
                <td>Pakistan Tehreek-e-Insaf was founded by Imran Khan on 25 April 1996
in Lahore. Founded initially as a sociopolitical movement, in June 1996 the first Central
Executive Committee of Pakistan Tehreek-e-Insaf was formed under the leadership of Imran
Khan, including Naeemul Haque, Ahsan Rasheed, Hafeez Khan, Mowahid Hussain, Mahmood Awan
and Nausherwan Burki as founding members. PTI began to grow slowly but did not achieve
immediate popularity. </td>
            </tr>
            <tr>
                <div class="PPP">
                    <td><img src="Images/PPP.png" class="Image"></td>
                    <td>Pakistan People Parties پارٹی پیپلز پاکستان</td>
                    <td>The Pakistan Peoples Party (PPP) was launched at its founding
convention held in Lahore on 30 November and 1 December 1967. At the same meeting,
Zulfiqar Ali Bhutto was elected as its chairman. Among the expressed goals of the party
were the establishment of an "egalitarian democracy" and the "application of socialistic
ideas to realize economic and social justice". A more immediate task was the struggle
against the dictatorship of General Ayub Khan, who was at the height of his power when
the PPP was formed.</td>
                </div>
            </tr>
            <div>
                <tr>
```

```
                    <td><img src="Images/PML-N.png" class="Image"></td>
                    <td>Pakistan Muslim League Nawaz پاکستان مسلم لیگ (ن</td>
                    <td>
                        The Pakistani Muslim League (Nawaz) (Urdu: پاکستان مسلم لیگ (ن),
romanized: Pākistān Muslam Lig (Nūn) abbr. PML(N) or PML-N) is a centre-right,
conservative political party in Pakistan. Alongside the Pakistan Tehreek-e-Insaf (PTI)
and Pakistan Peoples Party (PPP), it is one of the three major political parties of the
country. The party was founded by former Prime Minister Nawaz Sharif after the
dissolution of Islamic Democratic Alliance in 1993. The party's platform is generally
conservative, which involves supporting free markets, deregulation,lower taxes and
private ownership. Although the party historically supported social conservatism, in
recent years, the party’s political ideology and platform has become more liberal on
social and cultural issues.
                        One of several continuing factions of the original Muslim
League,the seeds of the party were sown following the 1985 Elections when the Prime
Minister of Pakistan Muhammad Khan Junejo organised the supporters of President Zia-ul-
Haq's dictatorship into a single party, known as the Pakistan Muslim League. After
President Zia's death in 1988, under the leadership of Fida Mohammad Khan, a large
faction split away from the Junejo-led Pakistan Muslim League, and formed a conservative
alliance with various right-wing and Islamist political parties, called the Islamic
Democratic Alliance. The alliance formed a government in 1990 under the leadership of
Nawaz Sharif.
                        In 1993, the alliance dissolved and the party assumed its current
shape, branding itself as the "Nawaz" faction of the Pakistan Muslim League, in contrast
to the "Junejo" faction.After its foundation, the PML-N, along with the People's Party,
dominated the two-party political system of Pakistan. However, after the 1999 coup, the
party was eclipsed by its own splinter faction, the Musharraf - backed Pakistan Muslim
League (Quaid), for almost a decade. PML-N regained popularity in the 2008 general
elections, when it was elected as the principal opposition party. It returned to power
following the elections of 2013, with Sharif elected as the Prime Minister for an
unprecedented third term. The party however faced a major setback following the
disqualification of Prime Minister Sharif in 2017. The situation was further made worse
when Sharif and his daughter Maryam were sentenced to imprisonment on charges of
corruption.
                        The party lost both the centre and the provincial government of
its stonghold Punjab,[17] to PTI in 2018 elections. As of 2020, it is the main opposition
party in the Parliament under the leadership of Sharif's younger brother Shehbaz.
                    </td>


                </tr>
            </div>

            <div>
                <tr>
                    <td><img src="Images/Jamaat-e-Islami%20Pakistan.png"
class="Image"></td>
                    <td>Jamaat-e-Islami Pakistan اسلام علمائے جمیعت </td>
                    <td>
                        Jamaat-e-Islami's founder and leader until 1972, was Abul A'la
Maududi, a widely read Islamist philosopher and political commentator, who wrote about
the role of Islam in South Asia.[18] His thought was influenced by many factors including
the Khilafat Movement; Mustafa Kemal Atatürk's ascension at the end of the Ottoman
Caliphate; and the impact of Indian Nationalism, the Indian National Congress and
Hinduism on Muslims in India. He supported what he called "Islamization from above",
through an Islamic state in which sovereignty would be exercised in the name of Allah and
Islamic law (sharia) would be implemented. Mawdudi believed politics was "an integral,
inseparable part of the Islamic faith, and that the Islamic state that Muslim political
action seeks to build" would not only be an act of piety but would also solve the many
(seemingly non-religious) social and economic problems that Muslims faced.
                        Jamaat-e-Islami Headquarter in Lahore
                        Maududi opposed British rule but also opposed the Muslim
nationalist movement (nationalism being un-Islamic) and their plan for a circumscribed
"Muslim state". Maududi agitating instead for an "Islamic state" covering the whole of
India[18]—this despite the fact Muslims made up only about one quarter of India's
population.
                        Jamaat-e-Islami thus actively opposed the partition of India,
with its leader Maulana Abul A'la Maududi arguing that concept violated the Islamic
doctrine of the ummah. The Jamaat-e-Islami saw the partition as creating a temporal
border that would divide Muslims from one another.
                    </td>
                </tr>
            </div>

            <div>

                <tr>
                    <td><img src="Images/Awami%20National%20Party%20.png"
class="Image"></td>
                    <td>Awami National Party پارٹی نیشنل عوامی</td>
                    <td>
                        Khan Abdul Wali Khan's political career had been built on the
tradition of intense Pashtun nationalism inherited from his father, Khan Abdul Ghaffar
Khan (Bacha Khan). Although it was formed after the partition of India, most of its
leaders had favoured a united India, and had opposed the partition of India and the
creation of Pakistan.[6] Most of the leaders were Indian National Congress right hands
before the partition of India. Both men were opposed to the creation of Pakistan, and
after the creation of Pakistan in 1947, they were imprisoned. In 1956 Wali Khan joined
the National Awami Party (NAP), led by a charismatic Bengali socialist, Abdul Hamid Khan
Bhashani. In 1965 the NAP split into two factions, with Wali Khan becoming president of
the pro-Moscow faction.[7] The party's members participated in 1970 parliamentary
elections through the Pakistan Peoples Party's platform and the National Awami Party,
forming a largest socialist alliance with Zulfikar Ali Bhutto in 1970. However, the
alliance fell apart and its members joined the Pakistan National Alliance.
                        In 1972 the party was strong enough to form coalition provincial
governments, with its partner the Jamiat Ulema-e-Islam (JUI) in Khyber Pakhtunkhwa and
Balochistan. These governments were short lived. Wali Khan was again jailed, and his
party was barred from politics when the Supreme Court upheld the finding of President
Bhutto that the NAP was conspiring against the state of Pakistan. General Zia-ul-Haq
subsequently withdrew the charges against the NAP. Wali Khan was released, joined the
National Democratic Party, and ultimately formed the Awami National Party. In the
meantime, Prime Minister Bhutto was imprisoned and executed in April 1979.
                    </td>
                </tr>
            </div>
        </table>

    </div>
</body>
</html>
```

Cast Vote Page:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Cast Vote</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link href="contactform.css" rel="stylesheet">
    <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet"
        integrity="sha384-
wvfXpqpZZVQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN"
crossorigin="anonymous">
    <script src="../../Scripts/bootstrap.js"></script>
    <script src="../../Scripts/jquery-3.3.1.js"></script>
    <link href="../../Content/bootstrap.css" rel="stylesheet" />


</head>
<style>
body{
       background-image: url("Image/cast_vote_background.jpg");
       background-repeat: no-repeat;
       background-position: center;
       background-size: cover;
       height: 100vh;
       width: 100vh;
}
h2{
       font-family: Helvetica ;
}
</style>
<body>

    <div class="wrapper">
        <h2> Vote Casting</h2>
<form action="#" method="POST">
            <div class="input-group">
                <div class="input-box">
                    <label>Party</label>
                    <input type="text" placeholder="Party" required class="name">
                    <i class="fa fa-user"></i>
                </div>
<div class="input-box">
                    <label>Post</label>
                    <input type="text" placeholder="MPA " required class="name">
                    <i class="fa fa-user"></i>
                </div>
</div>
<div class="input-group">
                <div class="input-box">
                    <label>Post</label>
                    <input type="text" placeholder="MPA" required class="name">
                    <i class="fa fa-user"></i>
                </div>
<div class="input-box">
```

```
                    <label>Email</label>
                    <input type="email" placeholder="Email" required class="name">
                    <i class="fa fa-envelope"></i>
                </div>
</div>

<div class="button-group">
                <div class="button-box">
                    <button type="submit"><i class="fa fa-paper-plane"></i> Cast
Vote</button>
                </div>
</div>


</form>
</div>
</body>
</html>
```

Controller:

Account Controller:

Account Controller is used to manage users accounts on a website.

The controller is decorated with the [Authorize] attribute so that only authenticated users can navigate through out the controller actions, except those actions which are decorated with [Allow Anonymous] attribute. [Allow Anonymous] attribute is only decorated to the actions where the unauthenticated and unauthorized uses can navigate such as Login, Register, External Logins (GET and POST actions) etc. where the same user cannot navigate to the LogOff() action as this is only for authenticated users.

Home Controller

Controller classes in AMVC are used to prepare the Model that will be mapped to a view for a particular resource.

Admin Login Controller

This controller is joined with the admin login table and there are a couple of jobs of this controller:

1) Check if the admin id is present or not, if yes then it moves toward the next function but if not then it will throw an alert message and that message will be show on interface.

2) Once one function is passed then the next job is to identify the correct email, If the email is correct then controller will pass to the next function, If the email is not correct then the controller will throw an alert on the webpage.

3) Once the email and Id is identified then the controller will check for the correct password, if the password is correct then you can login into the system but if not the you can't logged in to the system

User Login Controller

There are a couple of function that are used in this controller.

1) Login verification

For login, the controller will check the user ID and once the Id is found then the controller will check for correct email and if the email is correct then controller will check for the password if all of things are good then there will no alert message but if any of the verification is missed then alert message ill thrown.

2) Add User

The controller will check for no duplicate id and once there will be no duplicate id then user can be added into the database.

3) Delete user

User Id will have entered and if the id is present in the table then user data can be deleted.

4) Update User

User can also update the data using this function, user will provide an id and once the id is found the data can be updated.