

Online Pharmacy



PROJECT GROUP MEMBERS

Students Id	Student Name
Student1206801	Dawood Ali
Student1205939	Burhanuddin Ali
Student1205900	Sheharyar
Student1199222	Daniyal Shafique

CURRICULUM**6740****BATCH****1908F1****FACULTY****MS. SAMREEN RAFIQ****CO-ORDINATOR****MS. RIMSHA SABIR**

Logics

Home Controller

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using PharmaProject.Models;
using System.IO;

namespace PharmaProject.Controllers
{
    public class HomeController : Controller
    {
        private pharmadbEntities db = new pharmadbEntities();

        // GET: Home
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Gallery()
        {
            return View();
        }

        public ActionResult Create()
        {
            return View();
        }

        public ActionResult about ()
        {
            return View();
        }

        public ActionResult pipeline()
        {
            return View();
        }
    }
}
```

Job Controller

```

public ActionResult job()
{
    return View();
}
[HttpPost]
public ActionResult job( Job job)
{

    if (ModelState.IsValid)
    {
        string imagename = Path.GetFileNameWithoutExtension(job.Imageayi.FileName);
        string extension = Path.GetExtension(job.Imageayi.FileName);
        HttpPostedFileBase serverimage = job.Imageayi;
        int size = serverimage.ContentLength;
        if (extension.ToLower() == ".pdf" || extension.ToLower() == ".doc" ||
extension.ToLower() == ".docx")
        {
            if (size <= 32000000)
            {
                imagename = imagename + extension;
                job.Resume = "~/files/" + imagename;
                imagename = Path.Combine(Server.MapPath("~/files/"), imagename);
                job.Imageayi.SaveAs(imagename);
                db.Job.Add(job);
                db.SaveChanges();
                ModelState.Clear();
            }
            else
            {
                ViewBag.msg = "<script>alert('invalid error')</script>";
            }
        }
        else
        {
            ViewBag.msg = "<script>alert('Wrong Extension')</script>";
        }
    }

    return View();
}

public ActionResult joblist()
{
    return View(db.Job.ToList());
}

}
}

```

```
@model PharmaProject.Models.Job
```

```
@{
    ViewBag.Title = "job";
}
```

```
@using (Html.BeginForm("job", "Home", FormMethod.Post, new { @enctype = "multipart/form-data" }))
{
    @Html.AntiForgeryToken()

    <head>
        <!-- Required meta tags-->
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <meta name="description" content="Colorlib Templates">
        <meta name="author" content="Colorlib">
        <meta name="keywords" content="Colorlib Templates">
        <!-- Title Page-->
        <title>Apply for job by Colorlib</title>
        <!-- Font special for pages-->
        <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,800i"
rel="stylesheet">
        <!-- Main CSS-->
        <link href="~/job/css/main.css" rel="stylesheet" media="all">
    </head>

    <body>
        <div class="page-wrapper bg-dark p-t-100 p-b-50">
            <div class="wrapper wrapper--w900">
                <div class="card card-6">
                    <div class="card-heading">
                        <h2 class="title">Apply for job</h2>
                    </div>
                    <div class="card-body">
                        <form method="POST">
                            <div class="form-row">
                                <div class="name">Name</div>
                                <div class="value">
                                    @Html.EditorFor(model => model.Name, new { htmlAttributes = new {
@class = "input--style-6" } })
                                </div>
                            </div>
                            <div class="form-row">
                                <div class="name">Father Name</div>
                                <div class="value">
                                    @Html.EditorFor(model => model.Father_Name, new { htmlAttributes =
new { @class = "input--style-6" } })
                                </div>
                            </div>
                            <div class="form-row">
                                <div class="name">Email Address</div>
                                <div class="value">
                                    @Html.EditorFor(model => model.Email_Address, new { htmlAttributes
= new { @class = "input--style-6" } })
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </body>
}
```

```

        <div class="form-row">
            <div class="name">Contact</div>
            <div class="value">
                @Html.EditorFor(model => model.Phone_number, new { htmlAttributes
= new { @class = "input--style-6" } })
            </div>
        </div>
        <div class="form-row">
            <div class="name">City</div>
            <div class="value">
                @Html.EditorFor(model => model.City, new { htmlAttributes = new {
@class = "input--style-6" } })
            </div>
        </div>
        <div class="form-row">
            <div class="name">State</div>
            <div class="value">
                @Html.EditorFor(model => model.State, new { htmlAttributes = new {
@class = "input--style-6" } })
            </div>
        </div>
        <div class="form-row">
            <div class="name">Date of Birth</div>
            <div class="value">
                @Html.EditorFor(model => model.DOB, new { htmlAttributes = new {
@class = "input--style-6" } })
            </div>
        </div>
        <div class="form-row">
            <div class="name">Qualification</div>
            <div class="value">
                @Html.EditorFor(model => model.Qualification_, new {
htmlAttributes = new { @class = "input--style-6" } })
            </div>
        </div>
        <div class="form-row">
            <div class="name">Education</div>
            <div class="value">
                @Html.EditorFor(model => model.Education, new { htmlAttributes =
new { @class = "input--style-6" } })
            </div>
        </div>

        <div class="form-row">
            <div class="name">Upload CV</div>
            <div class="value">
                <div class="input-group js-input-file">
                    <input class="input-file" type="file" name="Imageayi">
                </div>
                <div class="label--desc">Upload your CV/Resume or any other
relevant file. Max file size 50 MB</div>
            </div>
        </div>
    </form>
</div>

    <div class="card-footer">
        <button class="btn btn--radius-2 btn--blue-2" type="submit">Send
Application</button>
    </div>

</div>
</div>
<!-- JQuery JS-->
<script src="~/job/vendor/jquery/jquery.min.js"></script>

<!-- Main JS-->
<script src="~/job/js/global.js"></script>

```

```
    </body>
  }

  <div>
    @Html.ActionLink("Back to List", "Index")
  </div>
```


Admin Controller

```
public class AdminController : Controller
{
    private pharmadbEntities db = new pharmadbEntities();
    // GET: Admin
    public ActionResult Index()
    {
        return View();
    }
    [HttpPost]
    public ActionResult Index(adlogin admin, string returnUrl)
    {
        var login = db.adlogin.Where(a => a.Username == admin.Username && a.Password ==
admin.Password).FirstOrDefault();
        if (login!=null)
        {
            FormsAuthentication.SetAuthCookie(login.Username, false);
            Session["admin"] = login.Username;
            if (ReturnUrl!=null)
            {
                return Redirect(ReturnUrl);
            }
            else
            {
                return RedirectToAction("Create", "Home");
            }
        }
        return View();
    }

    public ActionResult logout()
    {
        FormsAuthentication.SignOut();
        Session.Abandon();
        return RedirectToAction("Create", "Home");
    }
}
```

Capsules Controller

```

namespace PharmaProject.Controllers
{
    [Authorize]
    public class capsulesController : Controller
    {
        private pharmadbEntities db = new pharmadbEntities();

        // GET: capsules
        [AllowAnonymous]
        public ActionResult Index()
        {
            return View(db.capsule.ToList());
        }

        // GET: capsules/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            capsule capsule = db.capsule.Find(id);
            if (capsule == null)
            {
                return HttpNotFound();
            }
            return View(capsule);
        }

        // GET: capsules/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: capsules/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        public ActionResult Create(capsule capsule)
        {
            if (ModelState.IsValid)
            {
                string imagename = Path.GetFileNameWithoutExtension(capsule.Imageupload.FileName);
                string extension = Path.GetExtension(capsule.Imageupload.FileName);
                HttpPostedFileBase serverimage = capsule.Imageupload;
                int size = serverimage.ContentLength;
                if (extension.ToLower() == ".jpg" || extension.ToLower() == ".jpeg" ||
extension.ToLower() == ".png")
                {
                    if (size <= 2000000)
                    {
                        imagename = imagename + extension;
                        capsule.pic = "~/capsulefile/" + imagename;
                        imagename = Path.Combine(Server.MapPath("~/capsulefile/"), imagename);
                        capsule.Imageupload.SaveAs(imagename);
                        db.capsule.Add(capsule);
                    }
                }
            }
        }
    }
}

```

```

        db.SaveChanges();
        ModelState.Clear();
    }
    else
    {
        ViewBag.msg = "<script>alert('invalid error')</script>";
    }
}
else
{
    ViewBag.msg = "<script>alert('Wrong Extension')</script>";
}
}

return View();
}
// GET: capsules/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    capsule capsule = db.capsule.Find(id);
    if (capsule == null)
    {
        return HttpNotFound();
    }
    return View(capsule);
}

// POST: capsules/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"Id,Name,company_name,output,size,dimension,shipping_weight,Power,Price_")] capsule capsule)
{
    if (ModelState.IsValid)
    {
        db.Entry(capsule).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(capsule);
}

// GET: capsules/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    capsule capsule = db.capsule.Find(id);
    if (capsule == null)
    {
        return HttpNotFound();
    }
    return View(capsule);
}

// POST: capsules/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{

```

ONLINE PHARMACY

```
capsule capsule = db.capsule.Find(id);
db.capsule.Remove(capsule);
db.SaveChanges();
return RedirectToAction("Index");
}

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

Liquids Controller

```

namespace PharmaProject.Controllers
{
    [Authorize]
    public class LiquidsController : Controller
    {
        private pharmadbEntities db = new pharmadbEntities();

        // GET: Liquids
        [AllowAnonymous]
        public ActionResult Index()
        {
            return View(db.Liquid.ToList());
        }

        // GET: Liquids/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Liquid liquid = db.Liquid.Find(id);
            if (liquid == null)
            {
                return HttpNotFound();
            }
            return View(liquid);
        }

        // GET: Liquids/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Liquids/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        public ActionResult Create(Liquid liquid)
        {
            if (ModelState.IsValid)
            {
                string imagename = Path.GetFileNameWithoutExtension(liquid.Image2.FileName);
                string extension = Path.GetExtension(liquid.Image2.FileName);
                HttpPostedFileBase serverimage = liquid.Image2;
                int size = serverimage.ContentLength;
                if (extension.ToLower() == ".jpg" || extension.ToLower() == ".jpeg" ||
extension.ToLower() == ".png")
                {
                    if (size <= 2000000)
                    {
                        imagename = imagename + extension;
                        liquid.img = "~/liquidfiles/" + imagename;
                        imagename = Path.Combine(Server.MapPath("~/liquidfiles/"), imagename);
                        liquid.Image2.SaveAs(imagename);
                        db.Liquid.Add(liquid);
                        db.SaveChanges();
                        ModelState.Clear();
                    }
                }
            }
        }
    }
}

```

```

    }
    else
    {
        ViewBag.msg = "<script>alert('invalid error')</script>";
    }
}
else
{
    ViewBag.msg = "<script>alert('Wrong Extension')</script>";
}
}

return View();
}

// GET: Liquids/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Liquid liquid = db.Liquid.Find(id);
    if (liquid == null)
    {
        return HttpNotFound();
    }
    return View(liquid);
}

// POST: Liquids/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
    "Id,Name,Air_pressure,Air_volume,Filling_speed,Filling_range_,capacity_,Price")] Liquid liquid)
{
    if (ModelState.IsValid)
    {
        db.Entry(liquid).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(liquid);
}

// GET: Liquids/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Liquid liquid = db.Liquid.Find(id);
    if (liquid == null)
    {
        return HttpNotFound();
    }
    return View(liquid);
}

// POST: Liquids/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{

```

ONLINE PHARMACY

```
        Liquid liquid = db.Liquid.Find(id);
        db.Liquid.Remove(liquid);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

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

Tablets Controller

```

namespace PharmaProject.Controllers
{
    public class TabletsController : Controller
    {
        private pharmadbEntities db = new pharmadbEntities();

        // GET: Tables
        public ActionResult Index()
        {
            return View(db.Table.ToList());
        }

        // GET: Tables/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Table table = db.Table.Find(id);
            if (table == null)
            {
                return HttpNotFound();
            }
            return View(table);
        }

        // GET: Tables/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Tables/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind
to, for
        // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Id,Name,Comapny,Address,City,State,Postal,Country,Email,Phone,Comments")] Table table)
        {
            if (ModelState.IsValid)
            {
                db.Table.Add(table);
                db.SaveChanges();
                return RedirectToAction("Create");
            }

            return View(table);
        }

        // GET: Tables/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Table table = db.Table.Find(id);

```


ONLINE PHARMACY

```
        if (table == null)
        {
            return HttpNotFound();
        }
        return View(table);
    }

    // POST: Tables/Edit/5
    // To protect from overposting attacks, please enable the specific properties you want to bind
to, for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
    "Id,Name,Comapny,Address,City,State,Postal,Country,Email,Phone,Comments")] Table table)
    {
        if (ModelState.IsValid)
        {
            db.Entry(table).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(table);
    }

    // GET: Tables/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Table table = db.Table.Find(id);
        if (table == null)
        {
            return HttpNotFound();
        }
        return View(table);
    }

    // POST: Tables/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Table table = db.Table.Find(id);
        db.Table.Remove(table);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

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

Tables (quote us)

Controller

```

namespace PharmaProject.Controllers
{
    [Authorize]
    public class TabletsController : Controller
    {
        private pharmadbEntities db = new pharmadbEntities();

        // GET: Tablets
        [AllowAnonymous]
        public ActionResult Index()
        {
            return View(db.Tablets.ToList());
        }

        // GET: Tablets/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Tablets tablets = db.Tablets.Find(id);
            if (tablets == null)
            {
                return HttpNotFound();
            }
            return View(tablets);
        }

        // GET: Tablets/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Tablets/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        public ActionResult Create(Tablets tablets)
        {
            if (ModelState.IsValid)
            {
                string imagename = Path.GetFileNameWithoutExtension(tablets.Image3.FileName);
                string extension = Path.GetExtension(tablets.Image3.FileName);
                HttpPostedFileBase serverimage = tablets.Image3;
                int size = serverimage.ContentLength;
                if (extension.ToLower() == ".jpg" || extension.ToLower() == ".jpeg" ||
extension.ToLower() == ".png")
                {
                    if (size <= 2000000)

```

```

        {
            imagename = imagename + extension;
            tablets.photo = "~/tabletfiles/" + imagename;
            imagename = Path.Combine(Server.MapPath("~/tabletfiles/"), imagename);
            tablets.Image3.SaveAs(imagename);
            db.Tablets.Add(tablets);
            db.SaveChanges();
            ModelState.Clear();
        }
        else
        {
            ViewBag.msg = "<script>alert('invalid error')</script>";
        }
    }
    else
    {
        ViewBag.msg = "<script>alert('Wrong Extension')</script>";
    }
}

return View();
}

// GET: Tablets/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Tablets tablets = db.Tablets.Find(id);
    if (tablets == null)
    {
        return HttpNotFound();
    }
    return View(tablets);
}

// POST: Tablets/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"Id,Name,Model,Dies,Max_pressure,Max_diameter,Max_depth,Production_capacity,machine_size,net_weight,po
wer,Price_") Tablets tablets)
{
    if (ModelState.IsValid)
    {
        db.Entry(tablets).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(tablets);
}

// GET: Tablets/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Tablets tablets = db.Tablets.Find(id);
    if (tablets == null)
    {
        return HttpNotFound();
    }
}

```

ONLINE PHARMACY

```
    }
    return View(tablets);
}

// POST: Tablets/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Tablets tablets = db.Tablets.Find(id);
    db.Tablets.Remove(tablets);
    db.SaveChanges();
    return RedirectToAction("Index");
}

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

Contact Controller

```

namespace PharmaProject.Controllers
{
    public class ContactsController : Controller
    {
        private pharmadbEntities db = new pharmadbEntities();

        // GET: Contacts
        public ActionResult Index()
        {
            return View(db.Contact.ToList());
        }

        // GET: Contacts/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Contact contact = db.Contact.Find(id);
            if (contact == null)
            {
                return HttpNotFound();
            }
            return View(contact);
        }

        // GET: Contacts/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Contacts/Create
        // To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "Id,Name,Email,Contact1,Message")] Contact contact)
{
    if (ModelState.IsValid)
    {
        db.Contact.Add(contact);
        db.SaveChanges();
        return RedirectToAction("Create");
    }

    return View(contact);
}

// GET: Contacts/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Contact contact = db.Contact.Find(id);
    if (contact == null)
    {
        return HttpNotFound();
    }
}

```

ONLINE PHARMACY

```
    }
    return View(contact);
}

// POST: Contacts/Edit/5
// To protect from overposting attacks, please enable the specific properties you want to bind
to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "Id,Name,Email,Contact1,Message")] Contact contact)
{
    if (ModelState.IsValid)
    {
        db.Entry(contact).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(contact);
}

// GET: Contacts/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Contact contact = db.Contact.Find(id);
    if (contact == null)
    {
        return HttpNotFound();
    }
    return View(contact);
}

// POST: Contacts/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Contact contact = db.Contact.Find(id);
    db.Contact.Remove(contact);
    db.SaveChanges();
    return RedirectToAction("Index");
}

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

Table (AdLogin)

```
CREATE TABLE [dbo].[adlogin] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Username] VARCHAR (50) NULL,  
    [Password] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Table (Capsules)

```
CREATE TABLE [dbo].[capsule] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Name] VARCHAR (50) NULL,  
    [company_name] VARCHAR (50) NULL,  
    [output] VARCHAR (255) NULL,  
    [size] VARCHAR (50) NULL,  
    [dimension] VARCHAR (50) NULL,  
    [shipping_weight] VARCHAR (50) NULL,  
    [Power] VARCHAR (50) NULL,  
    [Price ] VARCHAR (50) NULL,  
    [pic] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```


Table (Liquids)

```
CREATE TABLE [dbo].[Liquid] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Name] VARCHAR (50) NULL,  
    [Air_pressure] VARCHAR (50) NULL,  
    [Air_volume] VARCHAR (50) NULL,  
    [Filling_speed] VARCHAR (50) NULL,  
    [Filling_range ] VARCHAR (50) NULL,  
    [capacity ] VARCHAR (50) NULL,  
    [Price] VARCHAR (50) NULL,  
    [img] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Table (Tablets)

```
CREATE TABLE [dbo].[Tablets] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Name] VARCHAR (50) NULL,  
    [Model] INT NULL,  
    [Dies] VARCHAR (50) NULL,  
    [Max_pressure] VARCHAR (50) NULL,  
    [Max_diameter] VARCHAR (50) NULL,  
    [Max_depth] VARCHAR (50) NULL,  
    [Production_capacity] VARCHAR (50) NULL,  
    [machine_size] VARCHAR (50) NULL,  
    [net_weight] VARCHAR (50) NULL,  
    [power] VARCHAR (50) NULL,  
    [Price ] VARCHAR (50) NULL,  
    [photo] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Table (Contact)

```
CREATE TABLE [dbo].[Contact] (  
    [Id]          INT          IDENTITY (1, 1) NOT NULL,  
    [Name]        VARCHAR (50) NULL,  
    [Email]       VARCHAR (50) NULL,  
    [Contact]     INT          NULL,  
    [Message]    VARCHAR (255) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Table (Job)

```
CREATE TABLE [dbo].[Job] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Name] VARCHAR (50) NULL,  
    [Father_Name] VARCHAR (50) NULL,  
    [Email_Address] VARCHAR (255) NULL,  
    [Phone_number] INT NULL,  
    [City] VARCHAR (50) NULL,  
    [State] VARCHAR (50) NULL,  
    [DOB] VARCHAR (50) NULL,  
    [Qualification ] VARCHAR (50) NULL,  
    [Education] VARCHAR (50) NULL,  
    [Resume] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

Table (Table (Quote))

```
CREATE TABLE [dbo].[Table] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Name] VARCHAR (50) NULL,  
    [Comapny] VARCHAR (50) NULL,  
    [Address] VARCHAR (50) NULL,  
    [City] VARCHAR (50) NULL,  
    [State] VARCHAR (50) NULL,  
    [Postal] VARCHAR (50) NULL,  
    [Country] VARCHAR (50) NULL,  
    [Email] VARCHAR (50) NULL,  
    [Phone] INT NULL,  
    [Comments] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```