

REPORT TITLE

2018

JULY 1

COMPANY NAME

Authored by: Your Name



Logo
Name

Table of Contents

Bug Model and DB context:	4
.....	4
Db context:	4
.....	4
.....	5
DB_config File:	5
.....	5
.....	5
Bug Controller:	6
.....	7
.....	8
.....	9
.....	9
.....	10
Bug Index Page:	10
.....	10
.....	11
Bug Create new status drop down	11
.....	11
Home page:	11
.....	12
'	12
Privacy page:	13
.....	14

Please note all this code is what I did and was not auto generated through MVC.

Bug Model and DB context:

```
18 references
...public enum BugStatus
...{
...    open,
...    inprogress,
...    resolved,
...    closed
...}

37 references
...public class Bug
...{
...    /// [PrimaryKey("BugId")]
...    17 references
...    public int BugId { get; set; }
...    21 references
...    public string Summary { get; set; }
...    21 references
...    public string Description { get; set; }
...    35 references
...    public string Status { get; set; } = "open"; // Open or Resolved
...    23 references
...    public int CreatedByUserId { get; set; }
...}
```

Db context:

```
4  //using System.Threading.Tasks;
5  using Microsoft.EntityFrameworkCore;
6  using Ticketing_System_Interview_Exam.Models;
7
8  namespace Ticketing_System_Interview_Exam.Data
9  {
10     8 references
11     public class Ticketing_System_Interview_ExamContext : DbContext
12     {
13         0 references
14         public Ticketing_System_Interview_ExamContext(DbContextOptions<Ticketing_Sy
15             : base(options)
16         {
17             9 references
18             public DbSet<Ticketing_System_Interview_Exam.Models.Bug> Bug { get; set; } =
19             0 references
20             public DbSet<Ticketing_System_Interview_Exam.Models.User> User { get; set; }
21
22             0 references
23             protected override void OnModelCreating(ModelBuilder modelBuilder)
24             {
25                 base.OnModelCreating(modelBuilder);
26             }
27         }
28     }
29 }
```

```

9 references
... public DbSet<Ticketing_System_Interview_Exam.Models.Bug> Bug { get; set; } =
0 references
... public DbSet<Ticketing_System_Interview_Exam.Models.User> User { get; set; }

0 references
... protected override void OnModelCreating(ModelBuilder modelBuilder)
... {
...     base.OnModelCreating(modelBuilder);

...     modelBuilder.Entity<Bug>().ToTable("Bug");
...     modelBuilder.Entity<User>().ToTable("User Table");

...     //Seed data
...     modelBuilder.Entity<Bug>().HasData(
...         new Bug { BugId = 1, Summary = "Sample Bug 1", Description = "This is a sample bug 1" },
...         new Bug { BugId = 2, Summary = "Sample Bug 2", Description = "This is a sample bug 2" },
...         new Bug { BugId = 3, Summary = "Sample Bug 3", Description = "This is a sample bug 3" },
...         new Bug { BugId = 4, Summary = "Sample Bug 4", Description = "This is a sample bug 4" }
...     );
... }

```

DB_config File:

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "Ticketing_System_Interview_ExamContext": "Server=(localdb)\\mssqllocaldb;Database=Ticketing_System_Interview_ExamContext;Trusted_Connection=True;MultipleActiveResultSets=true;Encrypt=True"
  }
}

```

```

Ticketing_System;Trusted_Connection=True;MultipleActiveResultSets=true;Encrypt=True"

```

Bug Controller:

```

... public class BugsController : Controller
... {
...     private readonly Ticketing_System_Interview_ExamContext _context;

...     0 references
...     public BugsController(Ticketing_System_Interview_ExamContext context)
...     {
...         _context = context;
...     }

...     /// <summary>
...     /// this Method is to convert Status enum to string
...     /// </summary>
...     /// <param name="context">this is the list of Bugs objects</param>
...     /// <returns>a list of bugs that status has been converted</returns>
...     2 references
...     private List<Bug> GetStatus(List<Bug> context)
...     {
...         List<Bug> bugs = new List<Bug>();
...
...         foreach (var item in context)
...         {
...             switch (item.Status)
...             {
...                 2 references
...                 private List<Bug> GetStatus(List<Bug> context)
...                 {
...                     List<Bug> bugs = new List<Bug>();
...
...                     foreach (var item in context)
...                     {
...                         switch (item.Status)
...                         {
...                             case "0":
...                             {
...                                 item.Status = BugStatus.open.ToString();
...                                 bugs.Add(item);
...                                 break;
...                             }
...                             case "1":
...                             {
...                                 item.Status = BugStatus.inprogress.ToString();
...                                 bugs.Add(item);
...                                 break;
...                             }
...                             case "2":
...                             {

```

```

... case "1":
... {
...     item.Status = BugStatus.inprogress.ToString();
...     bugs.Add(item);
...     break;
... }
... case "2":
... {
...     item.Status = BugStatus.resolved.ToString();
...     bugs.Add(item);
...     break;
... }
... case "3":
... {
...     item.Status = BugStatus.closed.ToString();
...     bugs.Add(item);
...     break;
... }
... }
... return bugs;
... }

```

```

... // GET: Bugs
3 references
public async Task<IActionResult> Index()
... {
...     var list = await _context.Bug.ToListAsync();
...     return View(GetStatus(list));
... }
... // GET: Bugs-filtered, still need to change code
0 references
public async Task<IActionResult> RD_Index()
... {
...     List<Bug> listOfBugs = _context.Bug.Where(j => j.Status.ToString() == "0"
...         || j.Status.ToString() == "1").ToList();
...     return View(GetStatus(listOfBugs));
... }
... // GET: Bugs/Details/5

```



```

...// GET: Bugs/Create
0 references
✓ public IActionResult Create()
...{
...    ViewBag.Statuses = Enum.GetValues(typeof(BugStatus))
...        .Cast<BugStatus>()
...        .Select(s => new SelectListItem
...            {
...                Text = s.ToString(),
...                Value = ((int)s).ToString()
...            }).ToList();
...
...    return View();
...}

```

```

...switch (bug.Status)
...{
...    case "0":
...    {
...        bugstatus = BugStatus.open;
...        break;
...    }
...    case "1":
...    {
...        bugstatus = BugStatus.inprogress;
...        break;
...    }
...    case "2":
...    {
...        bugstatus = BugStatus.resolved;
...        break;
...    }
...    case "3":
...    {
...        bugstatus = BugStatus.closed;
...        break;
...    }
...}

```

```

.....case "2":
.....{
.....    bugstatus = BugStatus.resolved;
.....    break;
.....}
.....case "3":
.....{
.....    bugstatus = BugStatus.closed;
.....    break;
.....}
.....}

.....List<SelectListItem> selectListItems = new List<SelectListItem>();
.....selectListItems.Add(new SelectListItem($"{bugstatus}", "0"));
.....selectListItems.Add(new SelectListItem($"{BugStatus.resolved}", "1"));

.....ViewBag.Statuses = selectListItems.ToList();
.....bug.Status = "0";
.....return View(bug);
.....}

```

Bug Index Page:

```

<style>
.....body {
.....    background-color: #333;
.....    color: #fff;
.....    font-family: Arial, sans-serif;
.....    padding: 20px;
.....}

.....h1 {
.....    color: #dc3545; /* Red color for heading */
.....}

......table {
.....    background-color: #444; /* Dark background for table */
.....    color: #fff; /* White text color */
.....}

.....table th {

```

```

...h1 {
...color: #dc3545; /* Red color for heading */
...}

...table {
...background-color: #444; /* Dark background for table */
...color: #fff; /* White text color */
...}

...table th {
...background-color: #dc3545; /* Red background for table header */
...color: #fff; /* White text color for table header */
...}

...table td, .table th {
...border: 1px solid #dc3545; /* Red border for table cells */
...}

...table a {
...color: #dc3545; /* Red color for links */
...}

```

Bug Create new status drop down

```

...</div>
...<div class="form-group">
...<label asp-for="Status" class="control-label"></label>
...<select asp-for="Status" class="form-control" asp-items="ViewBag.Statu:
...<option value="">Select Status</option>
...</select>
...<span asp-validation-for="Status" class="text-danger"></span>
...</div>
...<div class="form-group">

```

Home page:

```

<style>
  ...body {
    ...background-color: #333;
    ...color: #fff;
    ...}

  ...jumbotron {
    ...background-color: #dc3545; /*Red background for jumbotron*/
    ...color: #fff; /*White text*/
    ...}

  ...lead {
    ...color: #fff; /*White text for lead paragraphs*/
    ...}

  ...btn-primary {
    ...background-color: #dc3545; /*Red background for primary buttons*/
    ...border-color: #dc3545; /*Red border for primary buttons*/
    ...}

<div class="container">
  ...<div class="row justify-content-center">
  ...  ...<div class="col-md-8 text-center">
  ...    ...<div class="jumbotron">
  ...      ...<h1 class="display-4">Welcome</h1>
  ...      ...<p class="lead">
  ...        ...Welcome to the Demo of my Ticketing System for Bug Tracking!
  ...      ...</p>
  ...    ...</div>
  ...    ...<p>
  ...      ...I'm delighted to present my ticketing system designed to streamline bu
  ...    ...</p>
  ...    ...<p>
  ...      ...Thank you for joining me!
  ...    ...</p>
  ...    ...<a href="#" class="btn btn-primary">Learn More</a>
  ...  ...</div>
  ...</div>

```

Privacy page:

```

<h1>@ViewData["Title"]</h1>

<p>
    ... Privacy Policy
    ... Effective Date: [Add Date Here LOL]
</p>

<ol>
<li>
    ... <strong>Introduction</strong><br>
    ... Thank you for using our Ticketing System for Bug Tracking ("the System"). We are
    ... </li>
<li>
    ... <strong>Information We Collect</strong><br>
    ... <ul>
    ... <li>Personal Information: When you use our System, we may collect personal
    ... <li>Usage Data: We may collect information about how you interact with our
    ... </li>
    ... </ul>
    ... </li>
<li>
    ... <strong>Information We Collect</strong><br>
    ... <ul>
    ... <li>Personal Information: When you use our System, we may collect personal
    ... <li>Usage Data: We may collect information about how you interact with our
    ... </li>
    ... </ul>
    ... </li>
<li>
    ... <strong>How We Use Your Information</strong><br>
    ... We use the information we collect to:
    ... <ul>
    ... <li>Provide and maintain our System;</li>
    ... <li>Improve, personalize, and enhance your experience;</li>
    ... <li>Communicate with you, including responding to your inquiries and providing
    ... <li>Analyze usage trends and preferences.</li>
    ... </ul>
    ... </li>
<li>
    ... <strong>Sharing Your Information</strong><br>
    ... We may share your information:
    ... <ul>
    ... <li>With service providers who assist us in operating our System;</li>
    ... <li>When required by law or to protect our rights.</li>
    ... </ul>
    ... </li>

```

```

...</ul>
...</li>
✓...<li>
...<strong>Security of Your Information</strong><br>
...<li>We implement reasonable security measures to protect your information from unau
...</li>
✓...<li>
...<strong>Your Choices</strong><br>
...<li>You have the right to:
...<ul>
...<li>Access, correct, or delete your personal information;</li>
...<li>Opt-out of receiving promotional communications.</li>
...</ul>
...</li>
✓...<li>
...<strong>Changes to This Privacy Policy</strong><br>
...<li>We may update this Privacy Policy from time to time. We will notify you of any
...</li>
✓...<li>
...<strong>Contact Us</strong><br>
...<li>If you have any questions about this Privacy Policy, please contact us at [Con
...</li>
</ol>

```