



Internal Job Board

W. Michael Jagers | Full Stack Web Developer | July 22nd, 2021

fsdp.michaeljagers.net

/ About the Developer /

Michael Jagers resides in Northern Kansas City, MO with his wife, Jamie, and two children. As a developer in the industry for five years already, he sought the opportunity to hone his skills and increase his marketability through Centriq. Completing an accelerated course load on top of an already demanding 9-5, he learned valuable lessons in time management and opportunity cost. Nothing worth doing comes easy...

In his free time, he enjoys fitness, weightlifting and martial arts.

/ Project Synopsis /

The goal of this application was to create an easily accessible, mobile-responsive friendly way for current TechSavvy employees to search for and apply to internal job openings.

In order to apply, a resume is required and the ability to upload and manage their resume is available with an account.

TechSavvy Managers in the application will be able to open new positions in their location, view incoming applications, add notes, and ultimately accept or decline an applicant from a position.

Administrators will have the ability to add new locations as the company expands, as well as add additional positions to the pool for availability to managers.

/ Permissions /

Home	Index	About	Contact
Admin	X	X	X
Manager	X	X	X
Employee	X	X	X

Applications	Index	Details	Create	Edit	Delete	MyApplications
Admin	X	X				
Manager	X	X	X	X	X	
Employee						X

ApplicationStatus	Index	Details	Create	Edit	Delete
Admin	X	X	X	X	X

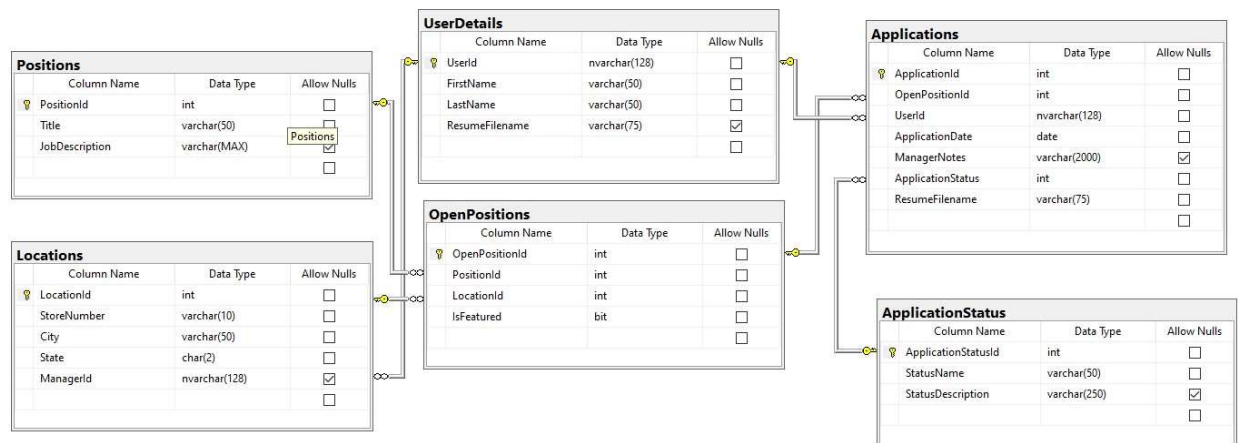
Locations	Index	Details	Create	Edit	Delete
Admin	X	X	X	X	X

OpenPositions	Index	Details	Create	Edit	Delete	FindJob	JobDetails
Admin	X	X	X	X	X	X	X
Manager	X	X	X	X	X	X	X
Employee						X	X

Positions	Index	Details	Create	Edit	Delete
Admin	X	X	X	X	X

Manage	UploadResume
Employee	X

/ ER Diagram /



Microsoft SQL was used as the database to store information on users, positions, locations, applications and more.

The entire project was version-controlled using git and is viewable at <https://github.com/michaeljaggers/JobBoard>

/ UI Highlights /

The application frankly speaks for itself and must be viewed in its entirety to fully appreciate its cohesiveness, however, there are a few things I would be happy to highlight for you to take note of when browsing:

- Index open-position mini list and featured jobs section
- Index section with live counter data on openings, resumes and user accounts
- Separate listings page from management with keyword search filtering and paging
- Separate job details page from management tailored for applicants

- A 'My Applications' page for employees to view current applications
- Altered edit functionality allowing only necessary fields for managers to ensure data integrity
- Custom field data created to make management of locations and positions easier
- One-click apply for employees only
- Workflow for applicants when a resume is not present

/ Code Snippets /

```
// GET: OpenPositions/FindJob
public ActionResult FindJob(string currentFilter, string searchString, int? page)
{
    int pageSize = 6; // Set 6 entries per page
    int pageNumber = (page ?? 1);

    if (searchString != null)
    {
        page = 1; // If a query is entered, start at page 1 of the new results
    }
    else
    {
        searchString = currentFilter; // Save the query into a filter to carry forward
    }

    ViewBag.CurrentFilter = searchString;

    // By default, return all open positions
    var openPositions = db.OpenPositions1.Include(o => o.Location).Include(o => o.Position)
        .OrderBy(o => o.Position.Title);

    if (!String.IsNullOrEmpty(searchString))
    {
        // Filter and return open positions containing keyword
        openPositions = openPositions.Where(o => o.Position.Title.Contains(searchString)
            || o.Location.City.Contains(searchString)
            || o.Location.State.Contains(searchString))
            .OrderBy(o => o.Position.Title);
    }

    return View(openPositions.ToPagedList(pageNumber, pageSize));
}
```

Keyword search filtering and paging.

```

// POST: Applications/Create
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Employee")]
public ActionResult Create(int openPositionId)
{
    Applications applications = new Applications();

    // Get current logged-in user
    var userId = User.Identity.GetUserId();
    // Get corresponding user details for user
    var userDeets = db.UserDetails1.FirstOrDefault(u => u.UserId == userId);
    string userResume = null;

    // Only employees will see the apply button, but we want to make sure we get data back
    if (userDeets != null)
    {
        userResume = userDeets.ResumeFilename; // Set their resume to the filename from the DB
    }

    // If above fails, this will remain null, otherwise set new application info
    // and save it to the database
    if (userResume != null)
    {
        applications.OpenPositionId = openPositionId;
        applications.UserId = User.Identity.GetUserId();
        applications.ApplicationDate = DateTime.Now;
        applications.ApplicationStatus = 3; // Pending
        applications.ResumeFilename = userResume;

        if (ModelState.IsValid)
        {
            db.Applications1.Add(applications);
            db.SaveChanges();

            // After a successful submission send use to application overview page
            return RedirectToAction("MyApplications");
        }
    }
    else
    {
        // Otherwise, send the user to the resume upload page to add their resume along
        // with a message
        Session["message"] = "A resume is necessary to apply for positions.";
        return RedirectToAction("UploadResume", "Manage");
    }

    return RedirectToAction("MyApplications");
}

```

One-click position application process with redirect.

W MICHAEL JAGGERS

michael.jaggers@gmail.com · 816-518-1498 · michaeljaggers.net
LinkedIn: linkedin.com/in/michaeljaggers/

FULL STACK DEVELOPER

.NET Framework
Full Stack Web Application Design
C# Fundamentals
Database Management

Unit Testing
Software Development Lifecycle
Agile Scrum Methodology
Continued Growth in IT Knowledge

TECHNICAL AND PROFESSIONAL SKILLS

Front End: HTML5, JavaScript, jQuery, jQueryUI, CSS3, Responsive/Mobile Web Development, Bootstrap, ReactJS

Middle Tier: Visual Studio, C#.NET, ASP.NET, LINQ, MVC, EF, Node.js

Back End: ADO.NET, SQL, SQL Server, SSMSE

Professional Skills: Version Control (Git), Adobe Photoshop, Troubleshooting, Critical Thinking, Communication, Project Management Fundamentals, Teamwork

INDEPENDENT DEVELOPMENT PROJECTS

- **Personal Site:** Created an MVC personal site to showcase abilities, projects, teammates, and allow for easy communication.
 - **StoreFront:** Created a secure application for managing product data. Application is built to simulate an online store front with a shopping cart. Administrators can manage product, category, and vendor data.
 - **S.A.T. Scheduling Administration Tool:** Created a secure application for managing product data. Application is built to simulate an online class scheduling system. Administrators will have the ability to manage students, courses, scheduled classes, and enrollments.
 - **Job Board:** Created a secure data driven ASP.NET MVC application from design through deployment for managing a company's open positions in various locations and gives employees the opportunity to apply for those therein.
-

TECHNICAL TRAINING

Full Stack Web Developer Program, Centriq Training
Kansas City, MO

01/2021 – 07/2021

- MVC Framework
- Trouble Shooting & Debugging
- Source Control
- Agile/Scrum
- Website Deployment
- Pair Programming
- Code Review

WORK EXPERIENCE

UI Engineer I, Dealer.com
Mission, KS

07/2021 – Present

- Develop Enterprise level user interfaces in React.js

Web Developer II, Dealer.com
Mission, KS

12/2017 – 07/2021

- Implement advanced changes to DDC sites using HTML5, XML, Velocity, CSS3 and JavaScript.
- Troubleshoot and research issues using browser developer tools.
- Write supporting software on an in-house platform called Gopher to automate production. tasks using JavaScript and GitHub, deployed to AWS through Jenkins.
- Code-review pull-requests into the Gopher GitHub repository.
- Implement custom designs from SOW mock-ups using the Dealer.com CMS platform.

Technical Project Manager, Dealer.com
Mission, KS

08/2016 – 12/2017

- Technical Project Manager for the Implementation Development team.
- Managed a queue of incoming work tasks.
- Vetted potential work as viable for the team.
- Acted as a resource on development related topics to Project Managers.

Senior Project Manager, Dealer.com
Mission, KS

07/2016 – 08/2016

- Senior-level Website Implementation contributor.
- Configured websites on the Dealer.com CMS platform to dealer specifications.
- Managed multiple ongoing site build projects simultaneously.

Front-End Designer I, VinSolutions
Mission, KS

09/2015 – 07/2016

- Designed, developed, and tested HTML and CSS that meet accessibility and browser standards.
- Utilized responsive design principles in desktop, mobile, and tablet environments.
- Interpreted existing website design and style guides by creating pixel-perfect web pages from mockups.

Website Tech Ops Specialist, VinSolutions
Mission, KS

09/2015 – 07/2016

- Performed 'Website Account Manager' responsibilities for all Lexus accounts.
- Managed the Salesforce client queue to ensure 24-hour turnaround on dealer requests.