

# DevChecks

## Task: Employee Addition Form

### Task Description:

You are tasked with creating a web pages for adding, viewing, updating, deleting employee details. The addition operation will be done via a form this form should include fields for the employee's name, job role, gender, whether it's their first appointment, start date, and any additional notes. The form will also have a save button to submit the data. After the add operation, all employees must be viewed in a grid (table), each table row is clickable on clicked it should preview the selected employee. And at the end there are options button which shows a menu with options for update or delete the record.

### Technologies to Use:

- **Frontend:** ASP.NET Core MVC or Razor Pages
- **Backend:** ASP.NET Core API
- **Database:** SQL Server
- **Language:** C#

### Requirements:

#### 1. Form Fields:

- **Employee Name (Text Input):** Allows input of the employee's name.
- **Job Role (Dropdown):** A dropdown list containing several predefined job roles. (Roles like Manager, Developer, Designer can be used for the dropdown values).
- **Gender (Radio Button):** Options for 'Male' and 'Female'.
- **First Appointment (Checkbox):** A checkbox that if checked indicates this is the employee's first appointment.
- **Start Date (Calendar):** A date picker to select the start date of employment.
- **Notes (Textarea):** A larger text field for any notes or additional information.
- **Save Button:** To submit the form data.

#### 2. Database:

- Create a SQL Server database with a table named **Employees** with columns corresponding to the form fields.

#### 3. Backend:

- Develop an API using ASP.NET Core that will handle the form submission. The API should validate the input data and then save it to the database.
- Ensure to handle exceptions and provide appropriate responses for both success and failure cases.

# DevChecks

## 4. Frontend:

- Create the form using ASP.NET Core MVC or Razor Pages.
- The form should be well-structured and user-friendly.
- Implement client-side validation to ensure data integrity before submission.

### Bonus Points:

- Implement server-side validation in the API.
- Use Entity Framework Core for data manipulation.
- Style the form using CSS or a framework like Bootstrap to improve the user experience.

### Draft Sketch of the Page:

Here's a basic wireframe for the form layout:

plaintextCopy code

```
+-----+
|                                     |
|                               Add Employee                               |
|                                     |
| Name:      [_____]              |
|                                     |
| Job:        [Dropdown: Developer ▼]  |
|                                     |
| Gender:     ( ) Male  ( ) Female    |
|                                     |
| [ ] First Appointment                |
|                                     |
| Start Date: [01/01/2022 ▼]          |
|                                     |
| Notes:                                           |
| [_____]                                         |
| [_____]                                         |
|                                     |
| [Save Button]                                  |
|                                     |
+-----+
```

### Instructions:

- Ensure the form is fully functional and data is properly captured and stored in the database.
- Consider security aspects, like preventing SQL injection and ensuring data validation both client and server-side.
- After completing the task, you should provide a demonstration to the team or submit your code for review.

# DevChecks

This task is designed to test your full-stack development skills, focusing on your ability to create a complete web application component from front to back. Good luck!

Please submit your completed task to **hr@devchecks.com** within a maximum of 3 days from today.