В заключение, разработанное веб-приложение представляет собой важный инструмент для компаний, облегчающий управление данными о персонале и оптимизирующий рабочие процессы. Его простой и интуитивно понятный интерфейс делает его доступным для широкого круга пользователей, а широкий спектр функциональных возможностей обеспечивает эффективное управление информацией о сотрудниках.

можно отметить, что разработанный класс EmployeeController играет ключевую роль в обработке запросов пользователя и взаимодействии с моделью данных и представлениями. Он содержит ряд методов, каждый из которых выполняет определенную функцию, обеспечивая функциональность приложения:

1. Конструктор принимает контекст базы данных ApplicationContext, обеспечивая доступ к операциям с базой данных.
2. Метод Error отображает представление с информацией об ошибке.
3. Метод Create отображает представление для создания нового сотрудника.
4. Метод Create (POST) обрабатывает POST-запрос с данными нового сотрудника и добавляет их в базу данных.
5. Метод GetEmployees получает список сотрудников с учетом фильтрации и сортировки и отображает их на странице.
6. Метод Delete обрабатывает POST-запрос для удаления сотрудника из базы данных.
7. Метод Edit отображает представление для редактирования информации о сотруднике.
8. Метод Edit (POST) обрабатывает POST-запрос с обновленными данными о сотруднике и сохраняет их в базе данных.
9. Метод UploadJsonFile обрабатывает POST-запрос для загрузки данных о сотрудниках из JSON-файла и добавляет их в базу данных.
10. Метод DownloadFile обрабатывает GET-запрос для скачивания JSON-файла с информацией о сотруднике.
11. Метод DownloadAllEmployees обрабатывает POST-запрос для скачивания JSON-файла с информацией о всех сотрудниках.

Кроме того, важным элементом являются классы SortState, SortHeaderTagHelper, SortViewModel и IndexViewModel, которые обеспечивают функциональность сортировки и фильтрации данных на странице, делая приложение более удобным и функциональным