./tomcat8/bin/startup.sh
mvn clean install
mvn tomcat7:redeploy

manager's: mazumisha@gmail.com Misha230411 github@gmail.com github@gmail.com

Тема моей сквозной работы, грузоперевозки. Показываю, как деплоим проект. Первый экран логина, мы можем выбрать залогиниться как менеджер или как водитель, введя его персональный номер. Заходим как менеджер. Наша задача, добавить нового водителя и новую фуру. Сначала добавим водителя: **Персональный номер: 567 Имя Mikhail Фамилия Mazurkevich адрес электронной почты driver567@gmail.com.** Показываем что водитель добавился и говорим что организован постраничный просмотр. Добавляем новый грузовик. **Номер: UP98723 Вместимость 50 тон. размер сменя 2 человека.** Так же мы узнали что один грузовик сломался и изменяем его ставим ему статус **BROKEN**. Теперь добавляем новый заказ. **Номер заказа 7 Груз номер 7 Wood 30 тонн Орел – Москва Груз номер 8 Bricks 20 тонн Москва-Тверь . Груз номер 9 Water 40 тонн Тверь-Ярославль. Груз 10 Apple 50 тонн Ярославль – Нижний Новгород. Выбираем фуру видим, что максимальный вес 50 тонн и выбираем фуру UP98723. Видим карту пути и продолжительность пути, следующий этап видим водителей и что нужно выбрать 2ух водителей. Как видим у всех разное время отработано, но она меньше 176 часов. Выберем нашего нового водителя 567 и у кого часов за 100 будет выбрать.** Видим нах оформленный заказ. И заходим в личный кабинет водителя 567 и видим что у него есть заказ и маршрут его и водителей и другую инфу.

Теперь про проект. Проект разделен на 2 модуля, модуль core, где содержатся сервисы, дао и ентити. Второй модуль web в котором JSP страницы и котроллеры.

В проекте реализовано Логгирование в файл, обработка ошибок, ошибки содержат код, описание которого содержится в error.properties. Реализован Фронт котнтроллер, фабрика контроллеров, фильтры EncodingFilter, CheckAccessFilter. JSP страницы используют библиотеку тегов JSTL. AppCOntext – singeltone который инжектит необходимы данные во все сервисы контроллеры и т .д.

Сервисы. Каждый сервис имеет описанный интерфейс, методы документированны и содержат коментарии. На сложную бизнес логику сервисов написаны тесты. В сервисы инжектятся необходимы ДАО классы.

Дао. Дао так же интерфейсы, реализован GenericDao, для базовых операций над всеми сущностями.

На этом все, спасибо за внимание.