Initial Thoughts / Steps

- Since a sample of car registration form is given (html file) I will start by opening the file to my browser and try to understand the requirements of the application. This will give a high level overview of the functionality etc.
- Check html file itself for retrieve functionalities, errors etc.
- Start creating a test plan, define testing areas (like functional, usability etc..).
- Try to be both influenced and not from the html file code, that is try to think (web) application as a black box and from the other side, taking into account html code (for example even if you know car plate regex validation, try to think cases to brake the field input)
- After testPlan, start to create specific test cases.Use Given/When/Then/Times format.Use specific prefix test case name (ex. TCF01 for functional, TCU-1 for usability related test etc.). Use priority.
- Start implementing tests automation. Start from priority1.
- Build a small compact testing framework related to web application, containing test management, test execution, reporting (summary, success, failure)
- Start with Maven, TestNG
- Implement DriverManager (controlling browser)
- Select a page object model for controlling interactions (actions) with web pages (pageObjects->single page actions, businessObjects->complex sequences of single actions, verifications)
- Find a simple solution for dependency injection of POs and BOs (GoogleGuice)
- Extend driver actions for frequent actions (secure click, selections, element attributes..)
- Create data (pojo) -> default random values
- Create a step for navigating to html sample files through browser
- Create Message/formValidation classes including related validation msg, values etc.
- Add Allure report
- Make custom testng listeners.
- Create configuration file (support)
- Logging
- Add instruction to README file
- Take a big breath… :)