# Final project

# Groups:

<https://docs.google.com/spreadsheets/d/1nt0mz1q6LuD8plqjPKvzX68qBvKjGWXXYkRhIhaRarU/edit#gid=0>

# Precondition

* Create a new account in Jira <https://www.atlassian.com/software/jira>
* Create a new account in Test rail <https://www.gurock.com/testrail>

**Note:** (You can use one of the temporary email services, for example <https://tempmailo.com> or <https://temp-mail.org/en/>)

**Hint:** You can create one account per group and share login details. This way all people will be contributing to the same project.

# Tasks

All tasks are separated into two categories: group tasks and individual tasks.

**Group tasks** – is a mandatory part. As a result, each group needs to prepare a presentation about work done, created list of test cases, found issues, risks analysis, etc. See more info below.

**Individual tasks** – can be done, if you’ve completed all group tasks and prepared group presentation.

## **Group tasks**

1. Application analysis
   1. Analyze the application assigned to your group (what type of application you have, what is the purpose of the application, what are the main sections/groups of functionalities that you can identify). You can do it individually as well as together as a group and then combine your observations.

Hint: if you identify main sections in the application, you can use it later during test cases creation

* 1. Identify all types of testing that you will be able to perform for this application (functional testing, accessibility, usability, exploratory, performance, etc……) You will need to perform each type of testing you specified at this moment.  
       
     Some hints:
     1. For Crossbrowser/crossplatform testing: test your site on different browserstack and devices, mobile/desktop and tablets (use your personal devices or free demo via saucelab or **browserstack** applications)
     2. Exploratory session is a technique which is executed after regular testing process.
     3. Feel free to use different tools for testing
  2. Create a list of test cases for the whole application. Add your test cases into Test management tool – Test rail (you can share 1 account, so that to add all test cases into 1 place).
  3. Perform test cases review for your group members (give your opinion on what can be improved)

1. Create a Test plan (use the example of the test plan and fill it with your data) – One test plan per whole group.
2. Execute functional tests, based on test cases created in 1.c task. Make comments, add screenshots as a proof of execution. You can do this in Test rail
3. Report found issues/bugs in Project management tool – Jira
4. Execute other types of testing, specified in 1.b and write down results.
5. Write down found issues and compile a list of recommendations for improvements.
6. Using SWOT Analysis technique analyze the risks that application has (Make SWOT table). See example here <https://www.javatpoint.com/swot-analysis-of-a-website> <https://venngage.com/blog/swot-analysis-templates/>
7. Prepare a presentation of the work done (you will be presenting your work for others groups)

**Group presentation:**

* Feel free to choose your own way to present your group result (google doc, google sheet, PowerPoint presentation, Prezi presentations, etc…)
* You can choose one speaker or give the word to every member of the group
* Presentation must include:
  + Demo of the application (Tell others what this application is for, show how it works). For example, you can demo the main functional flow.
  + List of types of testing you performed for your application
  + Test plan document (filled with your data)
  + Test cases, created in Test management tool with results of their execution
  + List of bugs/issues found
  + Also you can prepare pass/fail ratio char, based on test execution results
  + Result of other testing types execution/found issues/problems, list of recommendation what needs to be improved in the application
  + Results of SWOT risks analysis

**NOTE:**

Remember one of the testing principles: **Exhaustive testing is impossible**

This project puts you in condition, when you need to choose, how much testing is enough. If you start doing too much testing, too much details, you will not manage to finish testing during the planned time.

In real life we constantly have this situation, when we have limited time and need to decide, what we test first (most priority things) and what we skip (less priority things)

It is not a crime, if you didn’t test some parts of the application, but the most priority functionality needs to be tested.

## **Individual task**

If you feel like challenging yourself in test automation sphere:

1. Choose a simple test case from your group application
2. Using Selenium IDE record the steps of the test case and run it.
3. Export the test from Selenium IDE as Java code and analyze, what was generated

\*Advanced task

If previous tasks felt too easy ☺

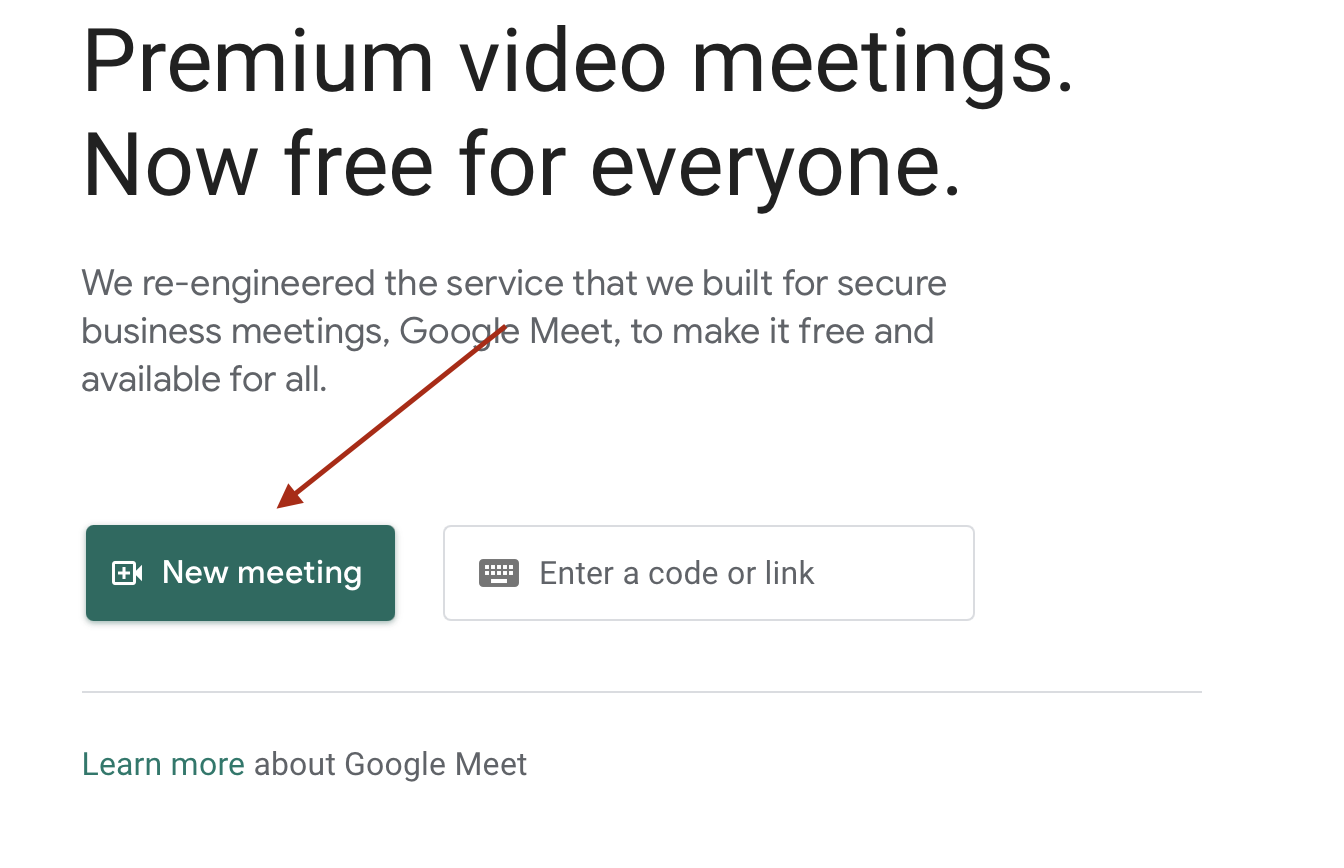
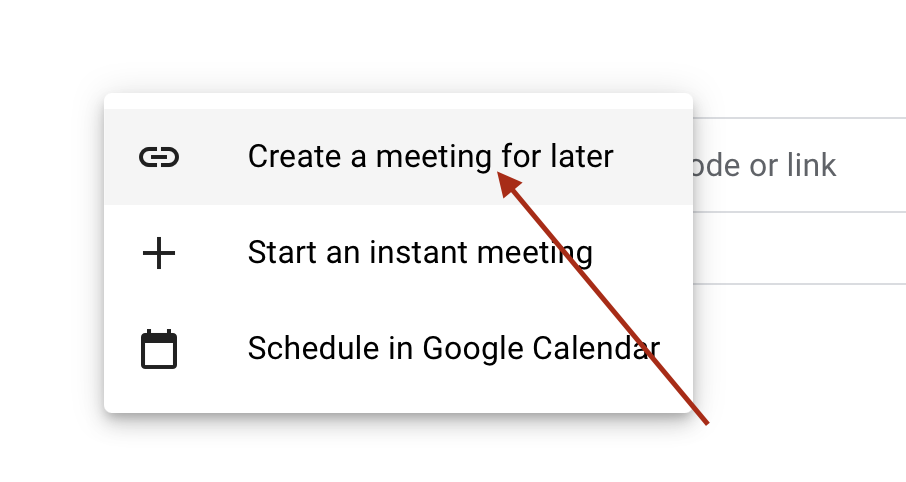
* Make the exported from Selenium IDE test run in IntelliJ
* Rewrite the **exported** code into the BDD approach (feature file, step definitions…)



And do not forget to rest ;)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# How to create a google meet meeting

* Go to <https://meet.google.com>
* Click New meeting button  
  
* Choose “Create a meeting for later”  
  
* Copy the meeting url and share it with your team members in Slack  
  