# 1. Question 1

It depends on different factors:

1. Requirements
    a. Waterfall is good for projects which have definite requirements and changes not at all expected.
    b. Agile is good for projects where requirements are expected to change and evolve.
2. Feature prioritization
    a. Agile: prioritization by value ensures the most valuable features are implemented first, thus reducing risk of having an unusable product once funding runs out.
    b. Waterfall ensures the customer gets everything they asked for. Increases risk of failure.
3. Customer involving
    a. Agile is a focused client process. So, it makes sure that the client is continuously involved during every stage.
    b. Waterfall requires customer involvement at milestones only.
4. Type of funding
    a. Agile works exceptionally well with Time & Materials or non-fixed funding. It may increase stress in fixed-price scenarios.
    b. Waterfall reduces risk in firm fixed price contracts by getting a risk agreement at the beginning of the process.
5. Testing
    a. Agile: testing is performed concurrently with software development.
    b. Waterfall: "Testing" phase comes after the "Build" phase. Hence, it has high chances of bugs to be found later in development where they are expensive to fix.

# 2. Question 2

1. I will record a screencast of the bug's reproduction.
2. I will report the full scenario from this screencast.
3. I will mention the bug's reproduction instability in the description.
4. I will collect and attach all possible components logs.

# 3. Question 3

Test flows:

1. Registration of a new user
    a. Positive
        i. Filling basic info -> Register -> Verify account by email.
    b. Negative
        i. Filling incorrect data -> Register -> Account isn’t registered.
2. Login using an existing account
    a. Positive
        i. Input credentials -> Sign in -> Success.
    b. Negative
        i. Input incorrect credentials -> Sign in -> Unsuccess.
3. Go through onboarding flow
    a. Positive

i. Check the main branch of the flow.
ii. Check all others branches of the flow.

b. Negative
i. Check the main branch of the flow using incorrect user actions (for example, incorrect data).
ii. Check all others branches of the flow using incorrect user actions.

# 4. Question 4

1. Identify successful controls.
2. Build a form data set.
3. Encode the form data set. The form data set is encoded according to the content type.
4. Submit the encoded form data set. The encoded data is sent to the processing agent designated by the *action* attribute using the protocol specified by the *method* attribute (get, post).
5. User agents render the response from the HTTP "get" and "post" transactions.

# 5. Question 5

```
select distinct c.Last_Name, c.First_Name
from Customer c
inner join Office o
on c.Office_id = o.Office_id
where o.Country = "Russia"
```

# 6. Question 6

## 6.1. [TC1] Listing search page via Home Page

For test project simplicity this test scenario contains general verifications only. It can be extended by various combinations.

| № | Step | Expected result | Comment |
|---|---|---|---|
| 1 | Open home page. | Home page is opened successfully. | |
| 2 | Check properties list display. | Properties list is displayed on the page. | |
| 3 | Check properties attributes. | All properties have name, price, address. | |
| 4 | Click the search button leaving all fields blank. | Search isn't performed. Warning message "Please select arrival and departure date" is displayed. | |
| 5 | Close the dialog by clicking "Close" button. | Dialog is closed. | |
| 6 | Save quantity Q1 of displayed properties. | Quantity is saved successfully. | |

| | | | |
|---|---|---|---|
| 7 | Open calendar by “Check-in” field. | Calendar is opened. | |
| 8 | Select first available date in calendar. | Date is selected successfully. | |
| 9 | Select the next date in the calendar. | Date is selected successfully. | |
| 10 | Check calendar presence. | Calendar is closed. | |
| 11 | Input value “2” into “Guests” field. | Value is inputed successfully. | |
| 12 | Click search button. | Search is completed. | |
| 13 | Check that search page is opened. | Search page is displayed. | |
| 14 | Check search results list. | Properties list is displayed. | |
| 15 | Repeat step 3. | | |
| 16 | Save quantity Q2 of displayed properties. | Quantity is saved successfully. | |
| 17 | Compare values of Q1 and Q2. | Values are different. | |

### 6.2. [TC2] Listing details page

For test project simplicity this test scenario contains general verifications only. It can be extended by various combinations.

| № | Step | Expected result | Comment |
|---|---|---|---|
| 1 | Open property detailed page. | Property page is opened successfully. | |
| 2 | Click “>” button on the current property’s photo. | Photo is changed to the next photo displayed in gallery thumbnails. | For test project simplicity we don’t consider the case when current photo has last position in thumbnails. |
| 3 | Click “<” button on the current property’s photo. | Photo is changed to the previous photo displayed in gallery thumbnails. | For test project simplicity we don’t consider the case when current photo has first position in thumbnails. |
| 4 | Check property attributes. | Property has name, address, description. | |
| 5 | Open calendar by “Check-in” | Calendar is opened. | |

| | | | |
|---|---|---|---|
| | field. | | |
| 6 | Choose a row with all free dates and select the first date. | Date is selected successfully. | For test project simplicity we don't take rows with blocked days. |
| 7 | Click on the date which is equal to the first date + 3. | Date is selected successfully. | For test project simplicity we don't analyze property's "minimum stay" quantity. Just use it as a customizable parameter equal to 3 by default. |
| 8 | Check calendar presence. | Calendar is closed. | |
| 9 | Note "Total" sum value as T1. | Value is noted. | |
| 10 | Input value "QAVACANCY2020" into "Coupon name" field. | Value is inputted successfully. | |
| 11 | Click "Apply" button. | Message "Coupon successfully applied!" is displayed. | |
| 12 | Verify coupon application. | Coupon name is added to payment details. | |
| 13 | Note "Total" sum value as T2. | Value is noted. | |
| 14 | Compare values of T1 and T2. | Values are different. | For test project simplicity we don't compare detailed payment values. |
| 15 | Click "Book now" button. | Booking details page is opened successfully. | |

### 6.3. [TC3] Booking details

For test project simplicity this test scenario contains general verifications only. It can be extended by various combinations.

Precondition: execute steps 1, 5-15 of "[TC2] Listing details page".

| № | Step | Expected result | Comment |
|---|---|---|---|
| 1 | Click on "First name" field and check it. | Field is editable. | |
| 2 | Press Tab button. | "First name is required" message is appeared. | |

| | | | |
|---|---|---|---|
| 3-4 | Repeat steps 1,2 for the “Last name” field. | | |
| 5-6 | Repeat steps 1,2 for the “E-mail address” field. | | |
| 7 | Check “Phone number” field. | Field is editable. | For test project simplicity we don’t verify that field isn’t required. |
| 8 | Try to input string value into it. | User can’t input string value. | |
| 9 | Check “Message to owner” field. | Field is editable. | For test project simplicity we don’t verify that field isn’t required. |
| 10 | Check “Card number” field. | Field is editable. | For test project simplicity we don’t verify that field isn’t required. |
| 11 | Check “Full Name” field. | Field is editable. | For test project simplicity we don’t verify that field isn’t required. |
| 12 | Check “Expiration date (Month)” list. | List isn’t empty. | For test project simplicity we don’t verify values of list. |
| 13 | Check “Expiration date (Year)” list. | List isn’t empty. | For test project simplicity we don’t verify values of list. |
| 14-15 | Repeat steps 1,2 for the “Billing address” field. | | |
| 16-17 | Repeat steps 1,2 for the “Zip code” field. | | |
| 18-19 | Repeat steps 1,2 for the “City” field. | | |
| 20 | Check “Billing country” list. | List isn’t empty. | |
| 21 | Click on “Billing country” field and press Tab button twice. | “Country is required” message is appeared. | |
| 22 | Note values for payment details as PD1. | Values are noted. | |
| 23 | Open calendar by “Check-out” field. | Calendar is opened. | |
| 24 | Click on date equal to check-out current date + 1. | Calendar is closed. | |

| | | | |
|---|---|---|---|
| 25 | Note values for payment details as PD2. | Values are noted. | |
| 26 | Compare values of PD1 and PD2. | Values are different. | |