Расписал свое тестовое задание ниже:

1. Я начал с исследования логики сайта и составления майндмэпа. На него ушло много времени. Вот ссылка: [майнд-мэп](https://miro.com/app/board/uXjVOrbLl5c=/?share_link_id=965222378900)
2. После майндмэпа я перешел к тест-кейсам: [тест-кейсы](https://docs.google.com/spreadsheets/u/0/d/1tgbClvkj0uE5kek-5nOhvCsf3tSyTeiSBRR0R5-lWJ4/edit). Покрыл далеко не весь функционал сайта, но старался выжать максимум к дедлайну. В тест-кейсах есть комментарии, где я объясняю логику написания некоторых проверок
3. Во время проектирования тестов я напоролся на баги, которые описал в файле: [описание багов](https://docs.google.com/document/d/1tVtBbj9UWAi_wIYrkrBbde-EqjW_-46axasKDxSfgCE/edit?usp=sharing).
4. Еще я использовал Pairwise для расчета комбинаций проверок на кнопку “Далее” страницы личных данных. Таблица из него лежит тут: [табличка](https://docs.google.com/spreadsheets/d/1L461VJ2jd4YM755PeBk9hws28Y2EavGc/edit?usp=sharing&ouid=109924859227011781400&rtpof=true&sd=true).

Про API нужно было расписать, что бы я делал с ним. В Charles я точно видел запрос к серверу с JSON на корзину, адрес и время доставки. Я бы попробовал дать серверу невалидные значения полей, запросы с не всеми обязательными полями, сломанные запросы, пустые запросы и запросы без необязательных полей. Если говорить про общий набор проверок для API, то я использую этот [чек-лист](https://gist.github.com/zeburek/8c165c9e8676945d75d91fe2f2addf8d). С доступом к документации я мог бы прогнать его через Postman и добавить специфичные запросы на логику сбердевайсов.

Это все. Возможно не стоило фиксировать логику в Miro, а сразу приступить к проектированию - так бы я написал больше тест-кейсов и покрытие было бы лучше. Но в любом случае, задание было классным. Спасибо за опыт и новые эмоции.