

Тестовое задание: Joomla + PHP (CRM stages + AI-driven development)

Цель: сделать прототип интерфейса и логики стадий CRM, где менеджер не может “перепрыгнуть” вперёд без выполнения обязательных действий.

Фокус оценки: корректность ограничений, простая архитектура, качество кода/тестов и зрелый AI-workflow.

1) Среда

- Joomla локально или Docker (на выбор).
- Реализация в виде компонента/страницы/модуля — на ваше усмотрение (обоснуйте в README).

2) UI: карточка компании

В карточке должны быть:

- текущая стадия,
- список доступных действий,
- блок “Инструкция/скрипт” для менеджера,
- история событий компании.

3) Логика стадий и событий

Стадии: Ice → Touched → Aware → Interested → demo_planned → Demo_done → Committed → Customer → Activated.

Условия переходов и ограничения реализовать согласно acceptance criteria (Given/When/Then)

Код	Название MLS	Ограничения (нельзя)	Вход (доступно если)	Выход (далее если)	Что есть
C0	Ice				
C1	Touched	Заведение заявки, отправка КП, планирование и показ демо	С компанией не было разговоров с ЛПР	Есть разговор с принимающим решение	Только кнопка для звонков. От отвеченного вызова — форма для комментария и для дискавери.
C2	Aware	Планирование и показ демо	Есть разговор с ЛПР	Есть заполненная форма для дискавери	Форма дискавери
W1	Interested	Заведение заявки, отправка КП	Если заполнена форма дискавери	Есть заполненная дата и время демонстрации	Кнопка планирования demo
W2	demo_planned	Заведение заявки, отправка КП	Есть дата плана demo	Проведено демо — зарегистрирован переход по ссылке	Кнопка проведения демо — по ссылке

W3	Demo_done		Есть лог того, что проведено демо < 60 дней	Есть заявка и/или счёт	Кнопка заведения заявки, кнопка отправки КП
H1	Committed		Есть счёт		
H2	Customer		Есть оплата		
A1	Activated		Есть одно удостоверение		
N0	Null				

События (минимум):

- попытка контакта,
- разговор с ЛПР + комментарий,
- заполнение дискавери,
- планирование демо (дата/время),
- демо проведено (кнопка/ссылка → событие),
- счёт выставлен,
- оплата получена,
- первое удостоверение выдано.

Допустимо мокировать интеграции, главное — чтобы логика была проверяемой.

4) Архитектура/БД

- Опишите модель данных (таблицы/сущности/индексы).
- Поясните, как бы вы масштабировали решение до ~10k компаний/день (принципы: индексация, журнал событий, минимизация блокировок, подход к логированию).

5) Тестирование

- Перечень unit-тестов (что покрываете и почему).
- Запуск тестов + лог прогона.
- Показать хотя бы один цикл “нашли баг → исправили → тесты зелёные” (коммитами или описанием).

6) AI-workflow (обязательная часть) Опишите:

- какие AI-инструменты использовали,
- как строили процесс (разбиение задач, промпты, проверка результата),
- как контролировали качество и риски (безопасность/галлюцинации/лицензии),
- где AI реально дал выигрыш (код/рефакторинг/тесты/документация/UX-тексты).

7) Что прислать

- Ссылку на Репозиторий,
- Ссылку на рабочий прототип с доступами,
- README: архитектура, модель данных, тесты, AI-workflow, что бы улучшили.