Используем PHP, MySQL (можно использовать любой фреймворк или слои для работы с базой, любые шаблонные движки и т.д.).

**Задание 1 (ООП и БД):**

Не нужно разрабатывать интерфейс для работы с юзерами или админку - всю информацию будем добавлять сразу в таблицы.

Сгенерировать и записать в БД две тысячи пользователей, у каждого пользователя случайным образом от 1 до 3-х номеров телефонов случайного оператора со случайным балансом от -50 до 150 грн.

Данные о пользователе (то, что должно быть на старте):

1. имя;
2. год рождения; (dd-mm-yyyy)
3. мобильный телефон (может быть несколько телефонов у одного пользователя - разные операторы, коды операторов: 50, 67, 63, 68); формат номера: код страны (380) + код оператора (2 цифры) + номер (7 цифр)
4. баланс на счету мобильного телефона (для каждого номера отдельный - может быть отрицательный и положительный - грн. и коп.).

Написать класс для работы с пользователями со следующей функциональностью:

1. зная ID пользователя получаем его имя, год рождения и список телефонных номеров;
2. возможность пополнить любой из номеров на сумму до 100грн. максимум за одно пополнение;
3. возможность добавить нового пользователя;
4. возможность добавить для пользователя номер мобильного телефона;
5. возможность удалить всю информацию о пользователе вместе с номерами телефонов;

Написать SQL-запросы которые будут выводить:

1. Баланс по каждому пользователю (сумма денег по всем номерам и операторам каждого пользователя)
2. количество номеров телефонов по операторам (список: код оператора, кол-во номеров этого оператора);
3. количество телефонов у каждого пользователя (список: имя пользователя, кол-во номеров у пользователя);
4. вывести имена 10 пользователей с максимальным балансом на счету (максимальный баланс по одному номеру);

Результат работы:

1. готовая БД;
2. скрипт, в котором реализован класс;
3. скрипт, в котором мы этот класс подключаем и вызываем его методы (в нем будем все проверять);
4. SQL-запросы, в отдельном файле (проверяем через Navicat, sqlyog);

Оцениваем:

* **внимательность и умение читать\понимать задание;**
* **умение сосредоточиться на главных моментах и не распыляться на реализацию, того, что не предусмотрено задачей;**
* **умение пользоваться поиском и мануалами;**
* **время, затраченное на выполнение задачи;**
* структуру базы данных;
* правильность составления запросов к БД;
* применение ООП на базовом уровне;

Каждая значимая часть кода (все методы, большие блоки условий) должны быть прокомментированы.