1. Поясните принципы разработки приложения в соответствии с паттерном MVC.



MVC — это шаблон программирования, который позволяет разделить логику приложения на три части:

* **Model (модель)**. Получает данные от контроллера, выполняет необходимые операции и передаёт их в вид.
* **View (вид или представление)**. Получает данные от модели и выводит их для пользователя.
* **Controller (контроллер)**. Обрабатывает действия пользователя, проверяет полученные данные и передаёт их модели.

Этот паттерн разработки нужен для того, чтобы разделить логические части приложения и создавать их отдельно друг от друга. То есть писать независимые блоки кода, которые можно как угодно менять, не затрагивая другие.

Например, чтобы можно было переписать способ обработки данных, не меняя при этом способ их отображения. Это позволяет эффективно работать нескольким программистам — каждый занимается своим компонентом. При этом разработчику не нужно вникать в чужой код и его действия никак не повлияют на другие фрагменты приложения.

1. Что описывает таблица маршрутизации?

Таблица, которая описывает соответствие URI

1. Что описывает таблица контроллеров?

**Controller (контроллер)**. Обрабатывает действия пользователя, проверяет полученные данные и передаёт их модели.

1. Поясните назначение маршрутизатора.

*Маршрутизация* определяет, как приложение отвечает на клиентский запрос к конкретному адресу (URI).

С помощью класса express.Router можно создавать модульные, монтируемые обработчики маршрутов. Экземпляр Router представляет собой комплексную систему промежуточных обработчиков и маршрутизации; по этой причине его часто называют “мини-приложением