

Отговор на въпросите

1. Какви платформи на онлайн магазини познавате?

Има изключително много онлайн платформи, които съществуват на днешния POST и GET са два често използвани метода в PHP за изпращане на данни между клиент (като уеб браузър) и сървър. Те се използват в HTTP (Hypertext Transfer Protocol) заявки, за да се извършат различни видове операции.

- GET метод:

GET методът се използва за заявка на данни от определен ресурс. Когато клиент изпраща GET заявка, данните се добавят към URL адреса като параметри във формата на ключ-стойност двойки. В PHP можете да достъпите данните, изпратени с GET метода, използвайки суперглобалната променлива `$_GET`.

Променливата `$_GET` е асоциативен масив, който съхранява параметрите и техните стойности. Това ви позволява да извличате и обработвате данните, изпратени в URL адреса.

GET заявките се използват обикновено за извличане на данни и не трябва да се използват за чувствителна или лична информация, тъй като данните са видими в URL адреса.

- POST метод:

POST методът се използва за изпращане на данни към сървъра за обработка. Когато клиент изпраща POST заявка, данните се включват в тялото на заявката, вместо да се намират в URL адреса. Този метод е по-сигурен от GET, тъй като данните не са директно видими в URL адреса.

В PHP можете да достъпите данните, изпратени с POST метода, чрез суперглобалната променлива `$_POST`.

Променливата `$_POST` също е асоциативен масив, който съхранява изпратените данни. Можете да ги извличате и обработвате чрез нея.

POST заявките се използват обикновено за изпращане на формуляри, качване на файлове и изпълнение на операции, които модифицират данни на сървъра

2. За какво се използва Bootstrap и jQuery?

Bootstrap се използва, заради тяхната голяма колекция от елементи. Някои от използваните елементи са въртележката в началната страница и също така долни колонтитул на всичките страници.

Докато jQuery се използва при създаването на страницата за регистриране. И по-точно в промяната от страница за влизане в профил, на страница за създаване на профил.

3. Използват ли се хеширани данни в разработката?

Не. В разработката на приложението, не се използват хеширани данни, макар че в последствие е препоръчително да се имплементира хешира на паролата на потребители, за да се запазва тяхната лична информация и също така за усилване на сигурността на приложението.

4. Как се свързва бекенда с фронтенда?

Използвайки променлива на име "\$rows", която в различните "if" проверки взима информация от базите данни. След това използвайки "for each" цикъл повтарям контейнерите, които със синтаксиса "\$row["x"]" изваждат нужните полета от всяка таблица и ги вкарват в изградените за тях контейнери.

5. Нормализирана ли е базата данни?

Структурата на базата данни е относително нормализирана. Таблиците "items" и "users" следват принципите на нормализацията и съдържат ключови полета. Всяка колона съдържа само едно атомарно поле и няма повтарящи се групи от данни. Това съответства на първа нормална форма (1NF) и втора нормална форма (2NF).

6. Как се подава заявката за логин на потребителя?

Използвайки гореспоменатия POST метод приложението преценява, дали ще изпълнява логин или регистрация. След като види че потребителя иска да си влезе в профила, до взима информацията за името и паролата на потребителя, като след това използвайки променливата "\$query" изпраща информацията на потребителя към базата данни, за да бъде проверена. След което се създава променлива на име "\$user", която използвайки метода "fetch()" взима нужната информацията на потребителя.

И последната стъпка е да се направи една "if" проверка, при която ако "\$user" дава "true" стойност, следователно се започва сесия и потребителя е изпратен в страницата на магазина. А ако стойността е "false" следователно се изписва грешката "Incorrect Username or Password !" и потребителя е накаран да въведе наново информацията на профила си.