Úvod

Dynamicky generované stránky = stránky, které mění svůj obsah bez nutnosti zasahovat do kódu. Pokaždé, když uživatel otevře stránku (jeho prohlížeč pošle GET Request)

HTTP reqeusty

Uživatel to sice nevidí, ale prohlížeč a server spolu komunikují pomocí tzv. HTTP requestů, které mohou používat různé metody. Ke každému requestu také přikládáme metadata (např. naše cookies nebo druh operačního systému)

Skladba URL Adresy

```
https://mole.lol/server.php?skola=SPSJed&trida=4B
```

Adresa URL se skládá z následujících částí:

- `https` - Protokol, přes který posíláme (pro weby vždy `http` nebo `https`, které používá SSL => je šifrované)
- `mole.lol` - Adresa serveru (v tomto případě doména, může být ale i IP adresa), specifikuje na jaký server posíláme request
- `server.php` - Adresa endpointu (koncového bodu) na serveru, na který posíláme žádost (v tomto případě php soubor)
- `?skola=SPSJed&trida=4B` - Parametry (argumenty) žádosti. V tomto případě posíláme 2 parametry (`skola` s hodnotou `SPSJed` a `trida` s hodnotou `4B`). Argumenty musí být poslední část URL a musí začínat `?` a být navzájem odděleny `&`.

HTTP Metody

GET

Slouží na získávání dat. Když uživatel zadá do adresního řádku prohlížeče URL adresu, prohlížeč pošle GET request a zobrazí HTML kód, který mu server vrátí jako odpověď. Správně by se neměl využívat pro zápis dat, ale technicky to lze. Parametry zadáváme přímo do URL adresy

POST

Slouží k zapisování dat. Prohlížeč ho pošle pouze při odeslání formuláře s `method="POST"` nebo pomocí JavaScriptu. Kromě URL a metadat posílá i `body` ve kterém můžeme poslat

argumenty jako JSON objekt.

Princip programování www stránek na straně serveru

= server side rendering (SSR, vykreslování na straně serveru). Funguje na principu generování HTML kódu stránky při každém requestu. Na stránce máme kód (např. v php), který na základě adresy a parametrů vygeneruje stránku. Běh kódu probíhá přímo na serveru, takže můžeme přistupovat i k tajným datům (např. údaje k databázi).

Popište jazyk PHP, jeho příkazy a funkce

PHP je specializový jazyk k tvorbě dynamických webových stránek. Lze ho psát funkcionálně i objektově. Do .php souboru lze psát HTML, CSS a JS, které se beze změny propíše do výsledné stránky. Poté můžeme mezi tagy `<?php` a `?>` vkládat PHP kód, který se spustí při každé žádosti na server. Všechny PHP příkazy musí být zakončeny středníkem.

Příklad PHP kódu

```php
<?php

$var = 1; // Vytvoření proměnné

// Výpis hodnoty
echo "Proměnná je $var";

// Smyčka for
for ($i = 0; $i < 5; $i++) {
        echo "Opakovačka $i";
}

// Smyška foraech
$zaci = array("Dan", "Max", "Kuba", "Lukáš");

foreach ($zaci as $zak) {
        echo $zak;
}

// funkce
function helloWorld($name) {
        echo "Hello $name";
}

helloWorld("Max");
```

```php
// Víceřádkové echo (vypsání formuláře)
echo <<<END

        <form action="" method="post">
                <input name="jmeno">
                <input type="submit">
        </form>

END;

// Reakce na odeslaný formulář
if (isset($_POST["jmeno"])) {
        echo "Jmenuješ se ".$_POST["jmeno"]; // Hodnota pole formulářem s
name="jmeno"
}

?>
```

Pro přístup k hodnotě z formuláře použijeme `$_GET` nebo `$_POST` dle použité metody, které jsou slovníky (dictionary) všech odeslaných hodnot. Proto pro hodnotu konkrétního pole použijeme `$_GET["jmeno"]`.

Databáze

```php
<?php

// Připojení k db
$mysqli = new mysqli("localhost", "name", "pass", "dbname");

// Error pokud nelze připojit
if ($mysqli->connect_error) {
        die("Připojení k databázi selhalo");
}

// Pošleme SQL příkaz
$res = $mysqli->query("SELECT * FROM zaci")

?>
```

Vypsání tabulky z DB

```php
<?php

$res = $mysqli->query("SELECT * FROM zaci")
```

```php
// Začneme HTML table
echo "<table>";

// Nebudeme rovnou vypisovat pro více možností
$header = "";

for ($i = 0; $i < $res->field_count; $i++) {
        // přidáme buňku do $header
        $header .= "<th>".$res->fetch_field_direct($i)->name."</th>";
}

// Vypíšeme $header
echo $header;

// Data tabulky
while ($zaznam = $res->fetch_array(MYSQLI_NUM)) {
        $row = "<tr>";

        for ($i; $i < $res->field_count) {
                $row .= "<td>".$zaznam[i]."</td>";
        }

        $row .= "</tr>";
        echo $row;
}

// Ukončíme tabulku
echo "</table>";

// Odpojíme se od DB
$mysqli->close();

?>
```

SQL Databáze

Jazyk pro interakci s databází. Různé databáze fungují různým způsobem, používají jiné technologie, ale lze s nimi komunikovat pomocí stejného jazyku.
SQL = Structured Query language
SQL můžeme psát přímo nebo pomocí nějaké knihovny/nástroje (phpMyAdmin, Prisma atd.)

Příklady SQL příkazů

```sql
-- Vytvoření tabulky
CREATE TABLE zaci {
        id INT PRIMARY KEY,
        jmeno VARCHAR(10),
        vek INT
}

-- Získání celé tabulky
SELECT * FROM zaci

-- Získání jen některých sloupců
SELECT (jmeno, vek) FROM zaci

-- Smazání dat celé tabulky
DELETE * FROM zaci

-- Vložení dat
INSERT INTO zaci (id, jmeno, vek) VALUES (1, "Josef Matuška" 36)

-- Filtrování
SELECT * FROM zaci WHERE vek = 18

-- Řazení
SELECT * FROM zaci ORDER BY vek ASC -- ASC = ascending, DESC = descending
```

Každá tabulka by měla mít primární klíč (PK), který by měl být unikátní. Pomocí něj můžeme spolehlivě vybrat jednotlivý řádek. Je doporučené u něj nastavit autoincrement (automatickou hodnotu).

MySQL

Open-source databázový software vytvořený firmou Oracle roku 1995. Je to nejznámější implementace SQL, ale v dnešní době je nahrazena modernějšími (převážně MariaDB a PostgreSQL)

Příklady použití DB

- Databáze hráčů a jejich předmětů
- Objednávky v E-shopu
- Známky žáka ve škole

Postup tvorby stránek

Je naprosto totožný s tvorbou stránek v HTML, CSS a JS, akorát musíme zvolit vhodné databázové schéma, protože měnit sloupce za běhu (tzv. migrace) je složité. Také si musíme dát pozor na útoky (převážně cross site scripting a SQL injection), které mohou nastat, pokud ukládáme a zobrazujeme data zadané uživatelem.