Metodický list pro učitele: Střední škola (SŠ)

Téma hodiny: Architektura aplikací, kyberbezpečnost REST API a integrace LLM pomocí MCP protokolu
Cílová skupina: SŠ / Gymnázia / Lycea (Předměty Informatika, Sítě, Programování)
Časová dotace: 45 – 90 minut

Cíle bloku:

1. Student umí analyzovat architekturu moderní webové aplikace (Frontend React, Backend NestJS, DB) a číst v ArchiMate diagramech.
2. Student teoreticky vysvětlí implementaci kyberbezpečnosti na aplikační vrstvě (autentizace pomocí JWT tokenů, RBAC a logické Guards).
3. Student objeví fungování Model Context Protocol (MCP) pro integraci umělé inteligence a řešení optimalizací v SQL (Indexy, princip Soft Delete).

Postup přípravy (před hodinou):

1. Na SŠ úrovni nebudou studenti pouze "klikat" na webu v roli uživatele, ale budou primárně **analyzovat technickou dokumentaci** softwaru. Poskytněte studentům přístup k vašim dokumentům: *„EduStack IS – ArchiMate diagramy“*, *„EduStack IS – Popis ArchiMate modelu“* a *„Database Index Analysis“* (jsou součástí příloh práce).
2. Dejte studentům přístup do testovací instance s vyšší rolí (např. *zástupce ředitele – DEPUTY*). Během úkolů mohou zkoumat síťovou HTTP komunikaci přes vývojářské nástroje v prohlížeči (klávesa F12 -> záložka Network -> struktura JSON odpovědí z REST API, zasílání tokenu).
3. Studenti pracují jako softwaroví architekti, snaží se rozklíčovat zadání v pracovních listech, na závěr své hypotézy společně diskutují nad přiloženým klíčem.

Vysvětlení pokročilých pojmů (Slovníček pro studenty):

- **JWT (JSON Web Token):** standardní zašifrovaný text, který backend server vydá klientovi po úspěšném přihlášení. Webový klient jej pak automaticky přikládá do HTTP hlavičky (`Authorization: Bearer <token>`) s každým dotazem. Token nese tzv. payload (identitu, školu, roli).
- **RBAC (Role-Based Access Control):** koncept bezpečnosti řízení přístupu. Práva k manipulaci s daty se nepřidělují konkrétní osobě (Pepa Novák), ale abstraktní roli (např. učitel, administrátor).
- **Guards (ve frameworku NestJS):** Middleware komponenta, která stojí jako nekompromisní "vyhazovač" před každým API endpointem. Z tokenu dekóduje roli uživatele a pokud nesouhlasí, okamžitě zablokuje dotaz do databáze (vrátí HTTP 403 Forbidden).
- **MCP (Model Context Protocol):** Nový open-source standard od společnosti Anthropic. Umožňuje AI modelům (LLM agentům) obousměrně a bezpečně komunikovat s lokálními firemními/školními databázemi pomocí malých skriptů, zvaných "Nástroje" (Tools). AI si tak nemusí data vymýšlet.

- **SSE (Server-Sent Events):** Architektura pro odesílání "streamovaných" zpráv jedním směrem ze serveru na klienta přes otevřené HTTP spojení (Na rozdíl od Pollingu nevyžaduje opakované dotazování klienta).
- **Soft Delete:** databázová operace. Řádek se v databázi nezničí příkazem `DELETE`, ale změní se u něj pouze časové razítko smazání (přidá se hodnota do sloupce `deletedAt`). Pro aplikaci se řádek tváří jako neexistující, ale v paměti zůstává kvůli integritě zachován.

Pracovní list 1: Datová vrstva a autorizace

Jméno: .....

Téma: Architektura REST API a ochrana zdrojů (Security)

Tvá mise: Stáváš se systémovým bezpečnostním analytikem. Budeš analyzovat bezpečnostní a návrhové prvky EduStack IS na základě dodané technické dokumentace a ArchiMate diagramů. Není důležité, jak tlačítka na obrazovce vypadají, ale jak je aplikace zadrátovaná uvnitř a chráněna před útočníky.

- ☐ **Krok 1 (Autentizační tokeny):** Systém využívá pro autentizaci formát JWT (JSON Web Tokens). Tabulka procesu *BP-01: Přihlášení a výběr kontextu* (v dokumentaci) popisuje v kroku 3 vydání `GLOBAL JWT` a v kroku 6, až následně po výběru konkrétní školy uživatelem vydání `TENANT JWT`. Zkus detailně vysvětlit technický smysl tohoto dvoufázového ověření z pohledu „Multi-tenancy“ architektury systému (to znamená architekturu, kdy na jednom backend serveru běží data desítek nezávislých škol naráz). Proč by v takovém IS pouhý jeden token způsobil bezpečnostní katastrofu?

  Analýza:

- [ ] **Krok 2 (Ochrana API přes Guards):** Backend aplikace je naprogramovaný ve frameworku NestJS a využívá middleware ochrannou vrstvu zvanou *Guards*. Představ si následující bezpečnostní incident: Velmi pokročilý žák obejde webové rozhraní aplikace a přes terminál nebo vývojářskou aplikaci Postman pošle manuálně sestavený HTTP POST požadavek přímo na skrytý API endpoint `/api/grading` se snahou zapsat si do SQL databáze jedničku z fyziky. Jak přesně a v jakých krocích backend server zjistí, že tento zápis do databáze nesmí provést, a vrátí žákovi stavový kód 403 Forbidden?

  Analýza:

- [ ] **Krok 3 (Princip oddělení moci):** Nahlédni do dokumentace – *Matice oprávnění* a popisu *Business vrsty*. Zjistíš zajímavou věc. Role Zástupce ředitele (`DEPUTY`) má v rámci webového IS prakticky totožná a maximální práva (může mazat i tvořit) jako samotný ředitel školy (`PRINCIPAL`). K jakému jedinému a velmi důležitému datovému modulu Zástupce přístup nikdy NEMÁ a proč je toto striktní oddělení obou rolí z hlediska interního IT auditu a kontroly řízení školy klíčové?

  Analýza:

Pracovní list 2: Umělá inteligence a MCP protokol

Jméno: .....

Téma: Integrace LLM agentů a Model Context Protocol

Tvá mise: Zanalyzovat backendovou integraci LLM (Large Language Models, např. od firem OpenAI či Anthropic). EduStack IS nevyužívá jen primitivní odesílání zpráv do chatu, ale implementuje inovativní strukturu MCP (Model Context Protocol), která umožňuje umělé inteligenci přímou, programatickou práci s databází a nástroji (aplikační vrstva AC-03).

- ☐ **Krok 1 (Agentní nástroje - Tool calling):** Nahlédni do dokumentace Aplikační vrstvy (Kapitola 11 o MCP Serveru). Oddělený Node.js MCP server obsahuje celkem 36 připravených programatických nástrojů. Kdyby správce systému s nejvyšším oprávněním napsal do AI chatu prompt: *"Vygeneruj mi kompletní falešná data pro novou testovací školu"* – AI model porozumí textu a místo odpovědi autonomně zavolá na serveru skripty, které databázi naplní daty. Vypiš přesné zdrojové anglické názvy alespoň 3 těchto MCP nástrojů (z kategorie *Seeding*), které agent pro naplnění tohoto složitého úkolu řetězově spustí:

- [ ] **Krok 2 (Architektura transportní sítě):** V ArchiMate diagramu *Technology Layer* a *Integrační pohled* vidíme, že hlavní Backend API komunikuje s MCP Serverem napřímo pomocí technologie **SSE (Server-Sent Events)** na portu `:3001`. Nepoužívá k tomu klasické blokující HTTP REST požadavky typu Request/Response (neboli tzv. HTTP Polling). Z jakého programátorského důvodu (z hlediska UX a fungování jazykových modelů) je pro konverzační AI agenty transport pomocí protokolu SSE mnohem vhodnější a efektivnější volba? *Analýza:*

- [ ] **Krok 3 (Zero Trust architektura a zranitelnost):** V technické dokumentaci o architektuře MCP Serveru se jasně píše následující zjištění: *"MCP server operuje přímo s databází zcela bez role-based autorizace."* (To z toho důvodu, že AI model nemá sám o sobě žádnou osobní identitu ani uživatelskou roli). Zformuluj největší kybernetickou zranitelnost tohoto architektonického rozhodnutí. Jakým způsobem a kde musí systémový administrátor (síťový architekt / DevOps) zabezpečit port `:3001` takového MCP serveru, aby školní data nedokázal snadno přes Prompt Injection napadnout a smazat zvenčí z internetu kdokoliv? *Analýza:*

Pracovní list 3: Export dat, reporty a databáze

Jméno: .....

Téma: Datová optimalizace SQL, tabulkové struktury a legislativa

Tvá mise: Moderní aplikace musí nejen správně fungovat na byznysové vrstvě, ale musí striktně plnit legislativu (GDPR) a být extrémně rychlé pod obrovskou zátěží dat (stovky tisíc záznamů v SQLite). Otestuj své systémové inženýrské myšlení při návrhu databáze.

- ☐ **Krok 1 (Data Lifecycle a Soft Delete):** Pro záměrné mazání uživatelských profilů (např. vyloučení problémového žáka nebo odchod učitele ze školy) systém EduStack IS nevyužívá fyzické smazání SQL záznamu z uživatelské tabulky (tzv. Hard delete příkazem `DELETE FROM Users WHERE...`), ale implementuje pod povrchem strategii **Soft delete**. Jak je tato úprava v tabulce SQL databáze technicky v praxi realizována (jaký typ sloupce se v návrhu schématu obvykle přidává)? Jaký má *soft delete* kritický význam z hlediska zachování tzv. **Cizích klíčů (Foreign Keys)** u historických známek? (Co by se rozbilo, kdyby se učitel, který rozdal stovky známek, natvrdo fyzicky smazal?) *Analýza:*

- [ ] **Krok 2 (Optimalizace výkonu a databázové indexy):** Nahlédni do technického dokumentu *Database Index Analysis*. Databázový inženýr v něm silně doporučuje vytvořit kompozitní (složený) SQL index nad obrovskou tabulkou `Attendance` (Docházka) primárně pro dvojici sloupců `(studentId, date)`. Zkus pod to napsat ukázku praktického (byť zjednodušeného) SQL dotazu `SELECT`, který by například generoval výpis absencí pro rodiče konkrétního žáka za měsíc květen, a u kterého databázový engine tento B-Tree index silně ocení a dotaz se výrazně zrychlí (protože se vyhne smrtelně pomalému prohledávání tabulky zvanému *Full Table Scan*). SQL Dotaz: `SELECT * FROM Attendance WHERE`

- [ ] **Krok 3 (BOM Encoding u exportů a parsérů):** Modul pro export osobních dat do GDPR generuje z API uživatelům datové soubory formátu `.csv`. Programátor backendu musel v kódu (v bytu bufferu) výslovně zajistit, aby se na úplný začátek každého generovaného souboru vložil speciální neviditelný bajtový znak, označovaný jako **BOM (Byte Order Mark)**. Jaký masivně rozšířený tabulkový software (na operačním systému Windows) si vyžádal tento programátorský "hack", bez kterého by v České republice docházelo např. u jména *Bedřich Štěpán* k chybnému rozkódování UTF-8 diakritiky (zničení znaků s háčky a čárkami)? *Analýza:*

Klíč a hluboké technické odpovědi (SŠ)

Tento list slouží výhradně učiteli pro vyhodnocení odpovědí studentů a pro řízení odborné "code-review" debaty v druhé polovině vyučovací hodiny.

Klíč k Pracovnímu listu 7 (Datová vrstva a autorizace)

- **Krok 1 (GLOBAL vs. TENANT JWT token):** EduStack je "Multi-tenant" systém (jedna centrální databáze na pozadí plynule obsluhuje mnoho nezávislých škol, které do sebe nevidí). První token, `GLOBAL JWT`, se vydává ihned po ověření kryptografie hesla a zaručuje backendu pouze fyzickou identitu (*„Potvrzuji, že tento e-mail a člověk existují“*). To ale pro operace nad API zdaleka nestačí. Tento člověk totiž může být na základní škole žákem, ale na vedlejší střední škole IT administrátorem. Teprve po zvolení konkrétní školy v prohlížeči systém zkontroluje tabulku členství a vydá kratší `TENANT JWT`, ve kterém je jako Payload zašifrován přesný kontext: *„Člověk A, Role v této škole: TEACHER, ID školy: 5“*. Backend pak propouští dotazy jen nad daty s TenantId=5 (prevence obrovského bezpečnostního problému zvaného *Tenant Bleeding*, úniku dat mezi školami).
- **Krok 2 (Ochrana API přes NestJS Guards):** Proces ochrany na straně Node.js serveru: **1.** HTTP POST požadavek z Postmanu obsahuje v hlavičce `Authorization: Bearer <token>`. **2.** Zásadní roli hraje *Guard* (middleware vrstva), který požadavek zachytí dřív, než se vůbec spustí business logika v Controlleru (a ochrání tak DB). **3.** Guard z JWT tokenu dekóduje atributy a najde roli `STUDENT`. **4.** Podívá se do kódu na endpoint `/api/grading`, u kterého je anotací/dekorátorem `@Roles('TEACHER', 'PRINCIPAL')` pevně stanoveno, kdo sem smí vstoupit. **5.** Guard vyhodnotí neshodu. Požadavek s okamžitou platností zahodí a vrací chybu `HTTP 403 Forbidden`. Žák z Postmanu s databází vůbec nepohne.
- **Krok 3 (Zástupce vs Audit Log - Oddělení moci):** Zástupce nemá z navigace (ani v API) přístup do modulu **Audit Log**. Jde o klasický bezpečnostní princip "Separation of Duties" (oddělení pravomocí). Ředitel nese před zákonem absolutní zodpovědnost za data. Audit log je neměnný revizní záznam každého kliknutí v celém IS. Kdyby měl do logu stejný přístup i zástupce s vysokými právy, mohl by (čistě teoreticky) zahlazovat stopy po svých vlastních chybách (smazaných známkách) či nepovolených modifikacích.

Klíč k Pracovnímu listu 8 (AI a MCP protokol)

- **Krok 1 (MCP Nástroje - Tool Calling):** Autonomní agent (LLM Model) si text přeloží do intentů a začne řetězově volat funkce backendu. Z kategorie Seeding využije pro stavbu školy např. nástroje: `seed_school_structure`, `seed_users`, `seed_teachers`, `seed_students`, a `seed_grades_and_schedule`. (Studentům stačí opsat z dokumentace - kapitoly 11. - jakékoliv z nich).
- **Krok 2 (SSE vs REST u konverzační AI):** Standardní HTTP REST funguje jako blokující "Request-Response" – klient pošle dotaz a čeká třeba 20 sekund, dokud server nezkompletuje celou odpověď. AI modely (LLM) ale generují text postupně po částech, po tzv. tokenech (tzv. streaming, píší jako na psacím stroji). SSE (Server-Sent Events) otevírá jednosměrný tunel přes obyčejné HTTP, kterým Node.js MCP server plynule "pushuje" zkompletovaná slova na Frontend ke klientovi v milisekundách. Klient díky tomu okamžitě čte, rozhraní „nezamrzne“ a není nutné systém zatěžovat špatnou technikou zvanou HTTP Polling (kde se klient ve smyčce neustále ptá "Už to máš hotové?").
- **Krok 3 (Bezpečnost MCP a síťová izolace):** Nástroje v MCP (např. `delete_school`) zapisují napřímo do SQLite databáze. Útočník by se mohl připojit na otevřený port MCP serveru a ovládnout data školy, protože v samotném MCP už žádný RBAC Guard není. Síťový architekt (DevOps) musí při nasazení zabezpečit, aby port MCP Serveru `:3001` běžel **striktně izolovaně na privátní uzavřené síti** (localhost / interní Docker síť), kam se z internetu nedá dostat. Přístup do tohoto portu smí mít výhradně ověřený Backend API server EduStacku (který ověřuje lidské tokeny), a teprve ten zprostředkovává proxy komunikaci pro LLM do MCP vrstvy. Zabraňuje se tak katastrofálnímu Prompt Injection útoku zvenčí.

Klíč k Pracovnímu listu 9 (Exporty a optimalizace databáze)

- **Krok 1 (Databázový Soft Delete):** V SQL návrhu tabulky se pro uživatele nepřepíše fyzicky paměť, ale přidá se nový sloupec, typicky označený jako `deletedAt` (datový typ TIMESTAMP). Při UI požadavku na smazání se řádek neničí, pouze se aktualizuje (`UPDATE Users SET deletedAt = NOW() WHERE id = 5`). Aplikační GET dotazy pak všude obsahují filtr `WHERE deletedAt IS NULL`. *Hlavní výhody:* relační integrita a archivace dat. Pokud bychom natvrdo udělali `DELETE FROM Users` na učitele, databáze by (kvůli cizím klíčům / Foreign Keys) spadla na chybu u všech známek a událostí v rozvrhu, které tento učitel kdy rozdal nebo vytvořil. Alternativně by se záznamy kaskádově smazaly (CASCADE), čímž by škola přišla o digitální historii a neprošla by auditem od inspekce MŠMT. Díky Soft Delete zůstane databáze konzistentní, i když se uživatel už do IS nepřihlásí.
- **Krok 2 (Využití B-Tree indexů):** Složený index srovná v databázovém stromě data podle kombinace žáka a datumu. Příklad zrychleného dotazu: `SELECT * FROM Attendance WHERE studentId = 154 AND date >= '2026-05-01' AND date <= '2026-05-31';` (Případně s klauzulí `BETWEEN`). Index databáze umožní pomocí algoritmu přeskočit miliony nevalidních záznamů celé školy za 5 let dozadu a okamžitě ukázat pouze jeden měsíc jednoho žáka. Odstraní se katastrofa zvaná *Full Table Scan*.
- **Krok 3 (BOM Encoding a Excel):** BOM (Byte Order Mark) odpovídá úvodním 3 bytům dokumentu (v hexadecimální podobě `EF BB BF`). Do kódu se vkládá specificky kvůli aplikaci **Microsoft Excel** na operačním systému Windows. Bez značky BOM Excel totiž při dvojkliku na stažený `.csv` soubor nedokáže automaticky zparsovat kódování UTF-8, otevře jej ve starém systémovém formátu ANSI (Windows-1250) a kompletně zkomolí veškerou českou diakritiku (např. *Bedřich Štěpán* se změní na rozsypaný čaj typu *BedÅ™ich Å tÄ›pÃ¡n*).