

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

WEBOVÁ APLIKACE PRO PODPORU VÝUKY  
INFORMATIKY NA 1. STUPNI ZŠ

Bakalářská práce

Autor práce: Olga Komžáková

Vedoucí práce: Mgr. Hana Vojáčková, Ph.D.

Jihlava 2026

# Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	<b>Olga Komžáková</b>
Studijní program:	Aplikovaná informatika
Garant studijního Programu:	Ing. Lenka Kuklišová Pavelková, Ph.D.
Název práce:	<b>Webová aplikace pro podporu výuky informatiky na 1. stupni ZŠ</b>
Vedoucí práce:	Mgr. Hana Vojáčková, Ph.D.
Cíl práce:	Cílem práce je navrhnout a implementovat webovou aplikaci, která podpoří výuku informatiky na 1. stupni základní školy. Aplikace bude obsahovat sborník témat a úloh v souladu s revidovaným Rámcovým vzdělávacím programem a organizační nástroje, jako jsou kalendář, ukazatel pokroku, evidence žáků a jejich hodnocení, editor úloh a export dat. Výsledkem práce bude funkční prototyp aplikace, který usnadní přípravu hodin a přispěje ke zvýšení efektivity výuky.

## Abstrakt

Bakalářská práce se zabývá návrhem a realizací výukové aplikace, jejímž cílem je zefektivnit výuku informatiky na prvním stupni základní školy. Práce se zaměřuje na klíčové oblasti ovlivňující vzdělávací proces, jako je vývoj dítěte, zákony učení, motivace, hodnocení, vliv médií, psychologie barev, design a požadavky rámcového vzdělávacího programu (RVP). Součástí práce je seznámení s frameworkem Symfony, který byl využit při implementaci aplikace. V závěru jsou zhodnoceny dosažené výsledky, přínosy práce a možnosti dalšího rozvoje aplikace.

## Klíčová slova

webová aplikace; UI; framework Symfony; MySQL

## Abstract

The bachelor's thesis focuses on the design and implementation of an educational application aimed at improving the teaching of informatics at the primary school level. The thesis addresses key aspects influencing the educational process, such as child development, learning principles, motivation, assessment, media influence, color psychology, design, and the requirements of the Framework Educational Program (RVP). The work also includes an introduction to the Symfony framework, which was used for the implementation of the application. The conclusion evaluates the achieved results, the contribution of the thesis, and possibilities for further development of the application.

## Keywords

web application; UI; Symfony framework; MySQL

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užití své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 15. března 2026

.....

Podpis studentky

## Poděkování

*Ráda bych poděkovala vedoucí bakalářské práce Mgr. Haně Vojáčkové, Ph.D. za odborné vedení, cenné rady a připomínky, které mi pomohly při jejím zpracování.*

*Děkuji všem vyučujícím za znalosti, které mi během studia předali, a za jejich trpělivost a podporu během celého studia.*

*Poděkování patří také mé kamarádce, která mi byla oporou a průvodkyní světem programování.*

*V neposlední řadě děkuji své rodině, především svým dětem, za jejich trpělivost, podporu a pochopení po celou dobu studia.*

## Obsah

<b>Seznam obrázků.....</b>	<b>7</b>
<b>Seznam zkratk.....</b>	<b>8</b>
<b>Úvod .....</b>	<b>9</b>
<b>1 Teoretická část .....</b>	<b>10</b>
1.1 Účel a cíl práce.....	10
1.2 Výuka informatiky.....	12
1.3 Použité technologie .....	15
1.4 Návrh uživatelského rozhraní aplikace .....	18
<b>2 Praktická část .....</b>	<b>20</b>
2.1 Struktura aplikace.....	20
2.2 Databáze .....	22
2.3 Vývoj aplikace .....	27
2.4 Testování aplikace .....	39
2.5 Nasazení aplikace.....	41
<b>3 Možnost dalšího rozšíření aplikace.....</b>	<b>43</b>
3.1 Rozšíření správy obsahu a poznámek.....	43
3.2 Interaktivita a optimalizace datového modelu.....	43
3.3 Monitoring a organizační funkce .....	44
3.4 Inkluze a škálovatelnost.....	44
<b>Závěr .....</b>	<b>45</b>

## Seznam obrázků

Obr. 1: Probíhající reformy kurikula.....	12
Obr. 2: Digitální kompetence učitele .....	13
Obr. 3: Struktura vzdělávací oblasti Informatiky .....	14
Obr. 4: Architektura MVC ve frameworku Symfony .....	16
Obr. 5: Ukázka rozšířeného diagramu entita–vztah (EER) .....	17
Obr. 6: Úvodní stránka výukové aplikace.....	18
Obr. 7: Příklad užití výukové aplikace.....	22
Obr. 8: ER model .....	24
Obr. 9: Relační model.....	27
Obr. 10: Dashboard žáka.....	29
Obr. 11: Dashboard učitele .....	30
Obr. 12: Dashboard koordinátora .....	30
Obr. 13: Editor lekcí.....	31
Obr. 14: Ukázka zpracování odkazu na video v JavaScriptu.....	32
Obr. 15: Editor kvízů.....	33
Obr. 16: Ukázka ukládání kvízu do databáze v kontroloru aplikace .....	33
Obr. 17: Ukázka generování seznamu žáků ve formuláři docházky.....	34
Obr. 18: Ukázka výpočtu souhrnných statistik docházky v šabloně Twig .....	35
Obr. 19: Ukázka implementace ukazatele pokroku v šabloně Twig .....	36
Obr. 20: Ukázka generování tabulky výsledků žáků v šabloně Twig .....	37
Obr. 21: Ukázka odeslání nové události kalendáře na server pomocí funkce fetch .....	38
Obr. 22: Vytvoření nového uživatele v roli žáka .....	38

## Seznam zkratk

CSS	Cascading Style Sheets
CSV	Comma-Separated Values
HTML	HyperText Markup Language
IT	Information Technology
MySQL	Structured Query Language
ORM	Object-Relational Mapping
PHP	Hypertext Preprocessor
RVP ZV	Rámcový vzdělávací program pro základní vzdělávání
ŠVP	Školní vzdělávací program
UI	User Interface
UX	User Experience
VŠPJ	Vysoká škola polytechnická Jihlava
www	World Wide Web
ZŠ	Základní škola

## Úvod

Informační technologie dnes hrají velmi důležitou roli nejen v každodenním životě, ale také ve vzdělávání. Proto se žáci již na prvním stupni základních škol učí pracovat s nejrůznějšími technologiemi a rozvíjet tak své digitální dovednosti. S rostoucím důrazem na digitální gramotnost je třeba hledat nové způsoby, jak výuku informatiky zatraktivnit a zároveň ji přizpůsobit současným potřebám školy i žáků.

Z vlastní zkušenosti, získané během desetileté praxe na prvním stupni základní školy, mohu potvrdit, že podmínky pro výuku informatiky nejsou vždy ideální. V mnoha případech chybí odborník, který by se výuce systematicky věnoval. Přestože se učitelé snaží žákům látku co nejlépe přiblížit, mnohdy narážejí na nedostatek zkušeností, což ve většině případů vede k tomu, že žáci ztrácejí zájem o práci s počítačem a informatiku vnímají spíše negativně.

Řešením by měla být webová aplikace, která učitelům i suplujícím pedagogům usnadní práci s přípravami a pomůže odbourat stres spojený s výukou informatiky. Cílem je vytvořit praktický nástroj, jenž poskytne přehledné a snadno dostupné výukové materiály, a tím umožní vést hodiny s větší jistotou a pozitivním přístupem. Žáci by díky tomu měli vnímat informatiku jako předmět, který je baví, motivuje a rozvíjí jejich schopnosti.

# 1 Teoretická část

Teoretická část je zaměřena na vývoj dítěte, význam informatiky ve vzdělávání, aktuální Rámcový vzdělávací program pro základní vzdělávání a přehled technologií, které budou využity při vývoji aplikace v praktické části.

## 1.1 Účel a cíl práce

Cílem práce je vytvořit webovou aplikaci, která sjednotí přístup k výuce informatiky na prvním stupni základní školy a nabídne pedagogům i zastupujícím učitelům efektivní podporu při přípravě vyučovacích hodin. Sjednocení přístupu spočívá zejména ve strukturovaném členění výukových hodin, jednotném vizuálním stylu a metodickém vedení učitele.

Práce je zaměřena na návrh, implementaci a testování aplikace v prostředí frameworku Symfony. Kromě něj budou využity také další nástroje a technologie, zejména databázový systém MySQL, programovací jazyk PHP, kaskádové styly a doplňkově jazyk JavaScript pro interaktivní prvky uživatelského rozhraní. Uvedené technologie umožní vytvořit moderní, responzivní a uživatelsky přívětivé prostředí, které bude snadno rozšiřitelné i pro budoucí potřeby školy.

Při návrhu a vývoji aplikace budou zohledněny vývojové potřeby dětí mladšího školního věku, jejich kognitivní schopnosti, principy učení a motivace, principy psychologie barev, význam výuky informatiky a požadavky aktuálního RVP. Důraz bude kladen na přehlednost, intuitivní ovládání a vizuální přitažlivost uživatelského rozhraní.

### 1.1.1 Vývoj dítěte mladšího školního věku

Mladší školní věk zahrnuje období přibližně od šesti do jedenácti let, tedy čas, kdy dítě vstupuje do školního prostředí a začíná si osvojovat základní školní dovednosti – čtení, psaní a počítání.

Děti mladšího školního věku jsou přirozeně zvědavé, aktivní a vnímavé. Mají silnou potřebu osvojovat si nové poznatky, ověřovat své schopnosti a získávat zpětnou vazbu (Vágnerová, 2000, s. 165). V tomto období je pro ně mimořádně důležitý pocit úspěchu – pozitivní hodnocení a pochvala posilují jejich sebedůvěru, vytrvalost a motivaci k dalšímu rozvoji. Naopak opakované neúspěchy mohou vést ke ztrátě zájmu o učení a k negativnímu postoji ke škole.

Z hlediska výuky informatiky je proto nezbytné, aby vzdělávací prostředí bylo přehledné, srozumitelné a motivačně působící. Výuková aplikace by měla podporovat aktivní zapojení žáků, poskytovat okamžitou zpětnou vazbu a přehledně vizualizovat dosažené výsledky. Zmíněné prvky přispívají k rozvoji vnitřní motivace žáků, umožňují sledovat jejich individuální pokrok a posilují pozitivní vztah k učení.

### 1.1.2 Kognitivní schopnosti

Jednou z významných složek kognitivního vývoje je senzomotorické učení. Jde o proces, při kterém je nutné využívat různé nástroje – například psací potřeby, hudební nástroje, počítačovou myš či klávesnici. Dané učení je poměrně náročné, protože vyžaduje propojení více mentálních a pohybových operací (Lemrová, 2024, s. 25).

**Proces učení obvykle probíhá ve třech fázích:**

1. *Kognitivní fáze* – představuje nutnost pochopení zadání. Vyžaduje soustředění, pozornost, paměť, řeč a myšlení.
2. *Výkonná fáze* – probíhá aktivní činnost, například při školní práci. Důležitou roli zde hraje zpětná vazba a práce s chybou.
3. *Fáze automatizace* – dochází k upevňování a opakování činností, automatizaci postupů a snižování únavy při provádění úkolů.

V kontextu výuky informatiky se zmíněné fáze projevují zejména při osvojování práce s digitálními technologiemi, kdy žáci postupně přecházejí od pochopení principů k samostatnému a automatizovanému používání jednotlivých nástrojů.

**1.1.3 Učení a motivace**

Učení představuje základní proces rozvoje dítěte, během něhož si osvojuje nové poznatky, dovednosti a způsoby myšlení. Mezi klíčové formy učení patří verbální a pojmové učení, které spolu úzce souvisejí a navzájem se doplňují.

Verbální učení se zaměřuje na práci s jazykem, tedy na osvojování pojmů, výrazů a jejich významů. Pojmové učení rozvíjí schopnost porozumět vztahům mezi jednotlivými jevy a systematizovat poznatky. Oba typy učení jsou pro vzdělávací proces zásadní, protože umožňují žákům vytvářet souvislosti mezi jednotlivými předměty a obory, například mezi přírodovědou, technikou a společenskými jevy (Lemrová, 2024, s. 27).

**Proces učení je ovlivněn několika základními zákony:**

1. *Zákon motivace* – zdůrazňuje význam vnitřní potřeby učit se a prožívat úspěch. Motivace podporuje aktivitu, soustředění a vytrvalost žáka.
2. *Zákon zpětné vazby* – spočívá v tom, že informace o chybě nebo úspěchu musí být žákovi poskytnuta co nejdříve, aby se předešlo upevnění nesprávného postupu.
3. *Zákon transferu* – vyjadřuje přenos zkušeností. Dovednosti a vědomosti získané při jednom typu učení mohou pozitivně ovlivnit zvládnání jiných úloh.
4. *Zákon opakování* – podporuje upevňování a automatizaci poznatků, přičemž je neúčinnější tehdy, pokud se kombinuje s motivací, zpětnou vazbou a přenosem zkušeností (Lemrová, 2024, s. 31).

Hodnocení je důležitou součástí vzdělávacího procesu, které žákovi poskytuje zpětnou vazbu o dosažených výsledcích a naplňování stanovených cílů. Kvalitní hodnocení by mělo reflektovat, v čem žák uspěl, v jakých oblastech se může zlepšit a jaký postup by měl následovat k odstranění případných nedostatků.

Z psychologického a pedagogického hlediska plní hodnocení několik základních funkcí, a to kontrolní, diagnostickou, regulační, motivační, výchovnou a informativní (Lemrová, 2024, s. 107). Pokud je hodnocení konstruktivní a srozumitelné, působí pozitivně na sebehodnocení žáků, podporuje jejich vnitřní motivaci a přispívá k dlouhodobému zájmu o učení.

## 1.2 Výuka informatiky

Výuka informatiky je na základních školách v České republice realizována na základě Rámcového vzdělávacího programu (dále jen RVP), který byl v roce 2021 aktualizován o nový vzdělávací obor Informatika (NPI, 2021).

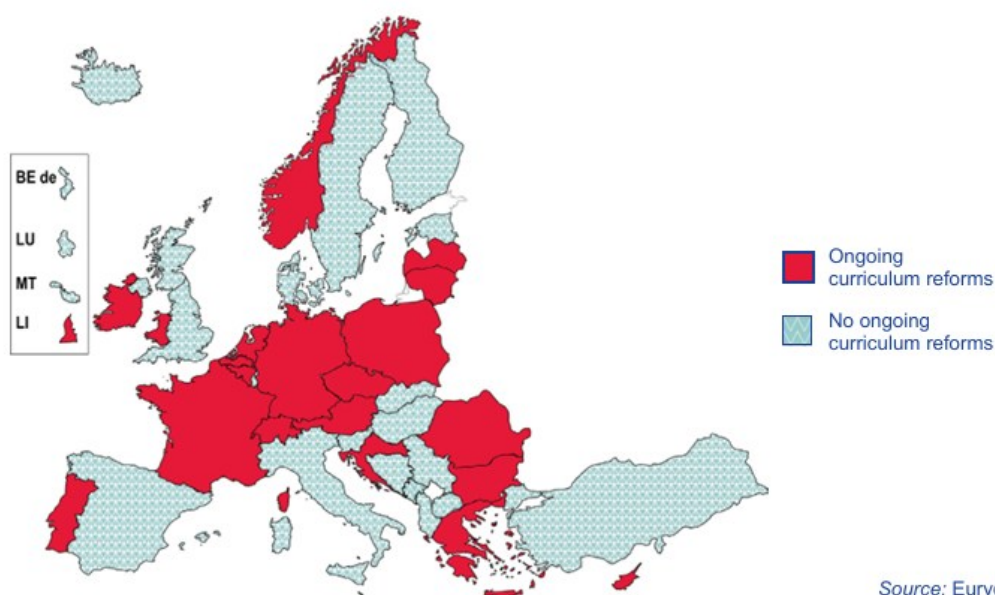
### 1.2.1 RVP

Rámcový vzdělávací program pro základní vzdělávání je závazný kurikulární dokument vymezující cíle, obsah a očekávané výstupy vzdělávání na základních školách v České republice. V oblasti informatiky klade důraz především na rozvoj informatického myšlení, práci s daty, algoritmizaci, porozumění principům fungování digitálních technologií a bezpečné chování v online prostředí.

Zásadní změnou je přechod od pouhého uživatelského ovládání zařízení k porozumění principů a k rozvoji tvůrčí práce s technologiemi. Výuka má vést žáky k pochopení, jak digitální technologie fungují, a k jejich smysluplnému používání v různých vzdělávacích situacích. Aktualizovaný RVP klade důraz i na mezioborové propojení a využití digitálních nástrojů napříč předměty (NPI, 2021).

### 1.2.2 Srovnání českého RVP s evropskými kurikulárními reformami

V evropském kontextu se česká reforma blíží trendům moderních kurikul, které mimo jiné také zdůrazňují informatické myšlení, digitální dovednosti a práci s daty. Podobný koncept uplatňuje například Estonsko, Finsko či Velká Británie. Studie Eurydice (2019) potvrzuje, že většina evropských zemí posiluje integraci informatiky do všeobecného vzdělávání a podporuje učitele v jejich profesním rozvoji. Rozsah a aktuálnost kurikulárních reforem v jednotlivých evropských zemích ilustruje obrázek 1.



Obr. 1: Probíhající reformy kurikula  
Zdroj: Eurydice (2019)

OECD (2023) zároveň upozorňuje, že úspěšnost zmíněných reforem úzce souvisí s podporou pedagogů, kvalitní metodikou a dostupností vybavení. Česká republika tak stojí před podobnými výzvami jako mnoho evropských států — tj. potřeba systematického vzdělávání učitelů, metodické podpory a modernizace školního prostředí.

### 1.2.3 Digitální kompetence učitele

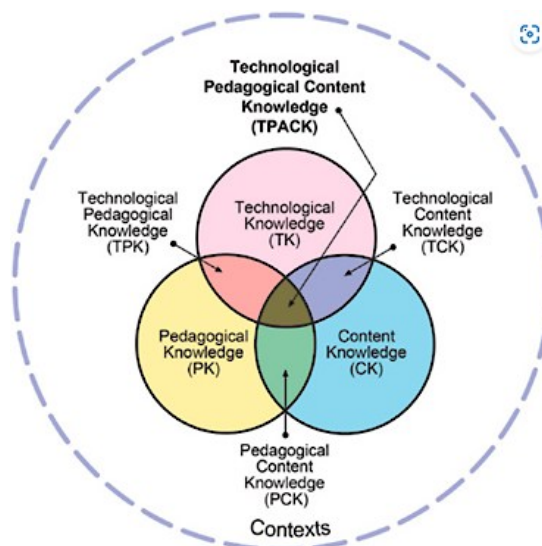
Kvalitní výuka informatiky je přímo podmíněna dostatečnou digitální kompetencí učitele. Podle výzkumů OECD (2023) a NPI (2024) mnoho učitelů uvádí, že mají omezené zkušenosti s implementací digitálních technologií do výuky nebo si nejsou jisti výběrem vhodných metod a nástrojů. Jedná se o skutečnost vytvářející tzv. *digitální propast na úrovni pedagogů*, kdy rozdíl ve znalostech a dovednostech ovlivňuje kvalitu vzdělávání.

**Digitální kompetence učitele zahrnuje zejména schopnost:**

1. vhodně vybírat a integrovat digitální nástroje do výuky,
2. podporovat rozvoj inforatického myšlení žáků,
3. zajišťovat bezpečné prostředí při práci v online aplikacích,
4. kriticky hodnotit digitální obsah a vést žáky ke stejné dovednosti.

Dostupné studie zároveň poukazují na přetrvávající potřebu dalšího vzdělávání učitelů, metodické podpory a sdílení dobré praxe, protože nedostatečná příprava může omezovat plné využití digitálních technologií ve výuce (Eurydice, 2019; Hlásná, Klímová & Paulová, 2017).

OECD zdůrazňuje, že digitální kompetence učitelů nelze chápat pouze jako technické ovládní digitálních nástrojů, ale jako komplexní soubor schopností, které zahrnují pedagogické rozhodování, práci s digitálním obsahem, podporu učení žáků a profesní rozvoj učitele v digitálním prostředí. Přehled hlavních oblastí digitálních kompetencí učitelů a jejich vzájemných vazeb znázorňuje obrázek 2 (OECD, 2023).



**Obr. 2: Digitální kompetence učitele**

Zdroj: OECD Digital Education Outlook 2023

### 1.2.4 Cíle výuky

Vzdělávací oblast *Informatika* je zaměřena především na rozvoj inforatického myšlení a porozumění základním principům fungování digitálních technologií. Výuka je postavena na aktivních činnostech, při nichž žáci řeší problémy, pracují s daty, navrhuji a ověřují postupy řešení a učí se rozhodovat, kdy je vhodné využít digitální technologie a kdy je účelné práci přenechat počítači.

Struktura vzdělávací oblasti Informatika je schematicky znázorněna na obrázku 3.



**Obr. 3: Struktura vzdělávací oblasti Informatiky**

*Zdroj: vlastní zpracování podle RVP ZV (NPI, 2021)*

Na prvním stupni základního vzdělávání si žáci prostřednictvím her, experimentování a praktických aktivit vytvářejí základní představy o datech, informacích a algoritmických postupech. Postupně se učí problém popsat, analyzovat a hledat jeho řešení, včetně ověřování jednoduchých algoritmů v programovacím prostředí. Nedílnou součástí výuky je také rozvoj bezpečného a odpovědného zacházení s digitálními technologiemi.

Na druhém stupni základního vzdělávání dochází k prohlubování získaných dovedností, zejména v oblasti systematického řešení problémů, práce s daty, automatizace postupů a pochopení principů kódování a modelování. Žáci se učí navrhovat, testovat a postupně vylepšovat technická řešení, zvažovat jejich dopady a rozvíjet kritické a etické myšlení v souvislosti s využíváním digitálních technologií.

Celkově výuka informatiky směřuje k tomu, aby žáci byli schopni digitální technologie nejen používat, ale také jim porozumět, kriticky je hodnotit a smysluplně je využívat v dalším vzdělávání i v běžném životě v digitální společnosti (NPI, 2021).

## 1.3 Použité technologie

Pro realizaci aplikace byla využita kombinace webových technologií, které zajišťují funkčnost, bezpečnost a uživatelskou přívětivost systému. Díky jejich vzájemnému propojení může aplikace sloužit jako praktický nástroj podporující výuku informatiky v běžném školním prostředí.

### 1.3.1 HTML

HTML (HyperText Markup Language) je základní značkovací jazyk používaný pro tvorbu webových stránek. Pomocí HTML lze definovat strukturu obsahu webových stránek, například nadpisy, odstavce, seznamy, tabulky nebo vložené multimediální prvky. HTML neřeší vizuální podobu stránky, ale poskytuje logické vymezení jednotlivých částí obsahu, které následně mohou být vzhledově upraveny pomocí stylů nebo doplněny o interaktivitu.

V kontextu webových aplikací HTML tvoří základní „kostru“ každé stránky. Díky své strukturované podobě umožňuje webovým prohlížečům správně interpretovat a zobrazit obsah uživateli. Současně je HTML klíčové pro přístupnost, protože umožňuje asistenčním technologiím (např. čtečkám obrazovky) rozpoznat význam jednotlivých částí stránky a interpretovat je pro uživatele s různými potřebami (W3Schools, 2025).

### 1.3.2 CSS

CSS (Cascading Style Sheets) slouží k vizuálnímu formátování obsahu vytvořeného v HTML. Umožňuje definovat barvy, rozložení, typografii či vzhled uživatelského rozhraní. Správně navržené CSS přispívá k přehlednosti aplikace a usnadňuje její údržbu. Moderní webové standardy zavádějí i responzivní design, který zajišťuje vhodné zobrazení na různých zařízeních (W3C, 2023).

### 1.3.3 JavaScript

JavaScript je skriptovací jazyk určený pro přidání interaktivity na webové stránky. Umožňuje dynamickou změnu obsahu, komunikaci se serverem, práci s formuláři nebo vytváření animací. Je základním pilířem moderního webového vývoje a jeho úlohu dále rozšiřují knihovny a frameworky. Pro tvorbu uživatelsky přívětivých aplikací je jeho použití nezbytné (W3Schools, 2025).

### 1.3.4 PHP

PHP (PHP: Hypertext Preprocessor) je serverový skriptovací jazyk určený především pro vývoj dynamických webových aplikací. Kód v jazyce PHP je zpracováván na straně serveru, kde probíhá aplikační logika, práce s databází a příprava dat pro jejich zobrazení uživateli. Výsledkem zpracování je následně generovaný HTML obsah, který je odeslán do webového prohlížeče.

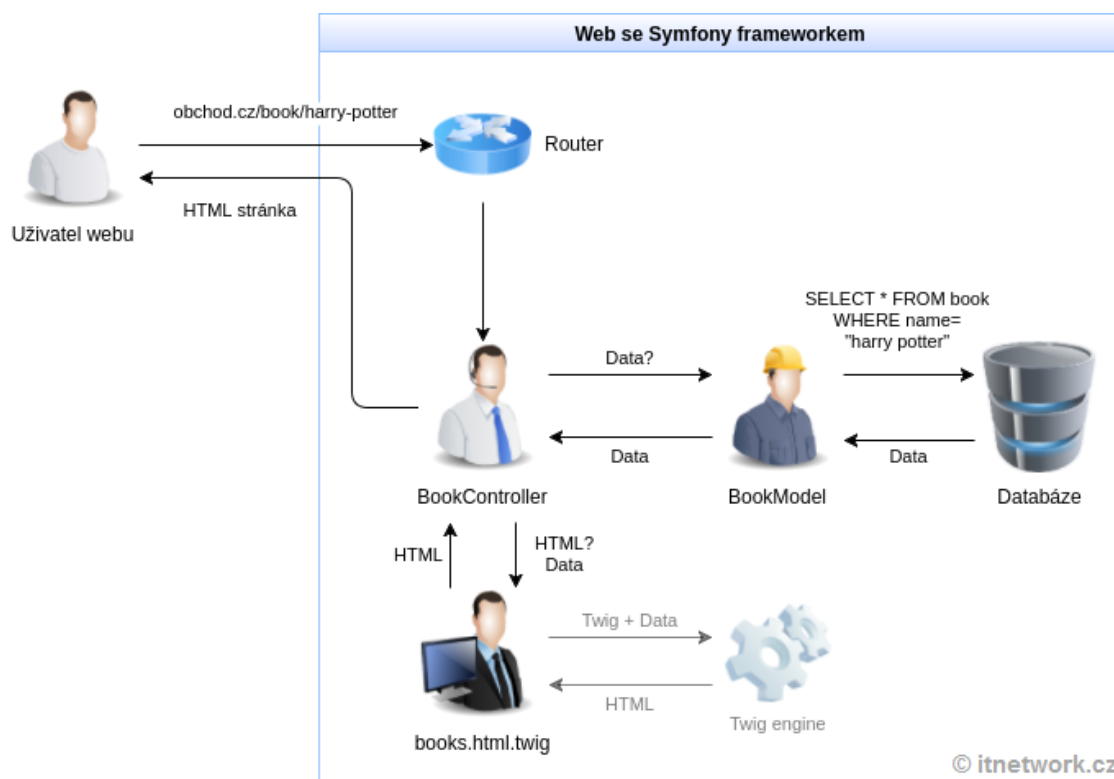
Jednou z hlavních výhod jazyka PHP je jeho jednoduchá integrace s databázovými systémy a široká podpora při vývoji webových aplikací. PHP umožňuje zpracování formulářů, práci s uživatelskými vstupy, správu uživatelských účtů a autentizaci. Jedná se o klíčové funkce moderních webových aplikací. Díky tomu je PHP vhodným nástrojem i pro tvorbu vzdělávacích aplikací, kde je nutné pracovat s různými rolami uživatelů, jako jsou žáci a učitelé.

V rámci práce slouží PHP jako základní programovací jazyk aplikační logiky a je úzce propojeno s frameworkem Symfony, který nad jazykem PHP poskytuje strukturované prostředí pro vývoj aplikace. Použití PHP v kombinaci s frameworkem přispívá k lepší přehlednosti kódu, vyšší bezpečnosti aplikace a snadnější údržbě i rozšiřování systému do budoucna. Jazyk PHP je dlouhodobě rozvíjen a jeho oficiální dokumentace poskytuje vývojářům rozsáhlou podporu (The PHP Group).

### 1.3.5 Symfony

Symfony je moderní PHP framework, který je založený na architektuře MVC (Model–iew– ontroller). Umožňuje přehlednou strukturu projektu, bezpečnou práci s databází prostřednictvím Doctrine ORM a efektivní správu rout, formulářů i uživatelských účtů (Stark, 2025). Výhodou Symfony je vysoká modularita, možnost opakovaného použití komponent a aktivní komunita. Díky tomu je vhodnou volbou pro tvorbu robustních webových aplikací včetně vzdělávacích platforem.

Základním principem architektury MVC je rozdělení aplikace do tří hlavních částí. Model zajišťuje práci s daty a komunikaci s databází, přičemž ve frameworku Symfony je patřičná vrstva nejčastěji realizována prostřednictvím nástroje Doctrine ORM. Controller slouží jako prostředník mezi uživatelem a aplikací – zpracovává požadavky, řídí aplikační logiku a rozhoduje o tom, jaká data budou předána k zobrazení. View je prezentační vrstva aplikace, která je ve frameworku Symfony realizována pomocí šablonového systému Twig a zajišťuje generování výsledného HTML obsahu.



Obr. 4: Architektura MVC ve frameworku Symfony

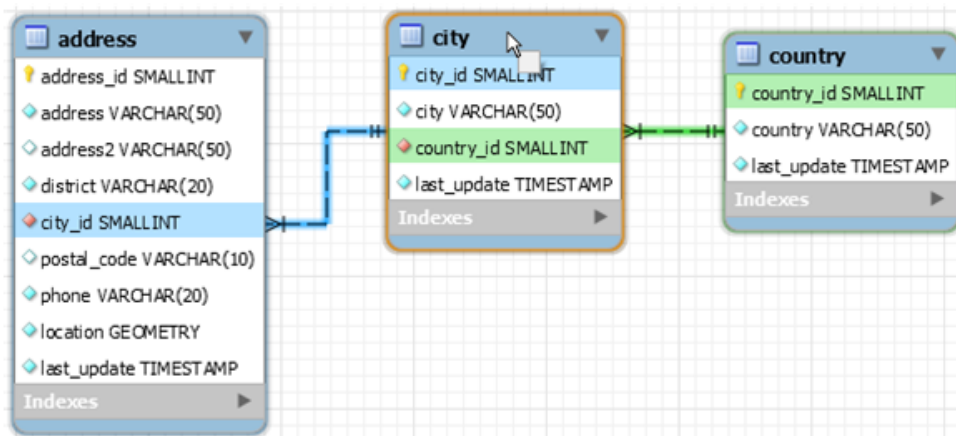
Zdroj: MÁČA, J., Itnetwork.cz

### 1.3.6 MySQL

MySQL je relační databázový systém určený k ukládání, správě a vyhledávání strukturovaných dat. Vychází z relačního databázového modelu, v němž jsou data organizována do tabulek propojených vzájemnými vztahy. Relační databázový model umožňuje systematickou správu dat, zajištění jejich konzistence a efektivní dotazování pomocí standardizovaného dotazovacího jazyka SQL (Structured Query Language).

Při návrhu databází, zejména u rozsáhlejších informačních systémů, je nezbytné věnovat pozornost databázovému návrhu ještě před samotnou tvorbou databázových tabulek. Návrhový proces obvykle začíná konceptuálním modelováním, jehož cílem je zachytit hlavní entity systému, jejich vlastnosti a vztahy mezi nimi. K uvedenému účelu se využívají zejména konceptuální nebo Entity-Relationship (ER) modely, které umožňují přehledně znázornit strukturu databáze nezávisle na konkrétním databázovém systému.

Na konceptuální úroveň navazuje logický, resp. relační model, ve kterém jsou jednotlivé entity převáděny na relační tabulky a jsou definovány primární a cizí klíče. Nedílnou součástí návrhu databáze je také proces normalizace dat, jehož cílem je minimalizace redundance, odstranění anomálií při vkládání, aktualizaci a mazání dat a zajištění jejich konzistence. Normalizace přispívá k přehlednější struktuře databáze a usnadňuje její dlouhodobou údržbu.



**Obr. 5: Ukázka rozšířeného diagramu entita-vztah (EER)**

*Zdroj: Oracle Corporation, MySQL Workbench Documentation*

Dalším důležitým aspektem návrhu relační databáze je stanovení integritních omezení, která zajišťují správnost a platnost uložených dat. Mezi základní integritní omezení patří zejména entitní integrita, referenční integrita a doménová omezení. Daná pravidla pomáhají předcházet nekonzistentním stavům databáze a podporují správnou funkci celého informačního systému.

Díky podpoře relačního modelu, standardu SQL a mechanismů pro práci s integritou dat poskytuje MySQL vhodné prostředí pro realizaci databází webových aplikací i rozsáhlejších informačních systémů. Jeho architektura umožňuje postupný přechod od databázového modelování k implementaci databázové struktury a následné práci s daty v aplikační vrstvě (Oracle Corporation, 2024).

## 1.4 Návrh uživatelského rozhraní aplikace

Návrh uživatelského rozhraní aplikace (dále jen UI) vychází z principů jednoduchosti, přehlednosti a intuitivního ovládání v souladu se zásadami uživatelské zkušenosti, aby se v aplikaci snadno orientovali jak žáci, tak učitelé.

### 1.4.1 Rozložení prvků a použitelnost

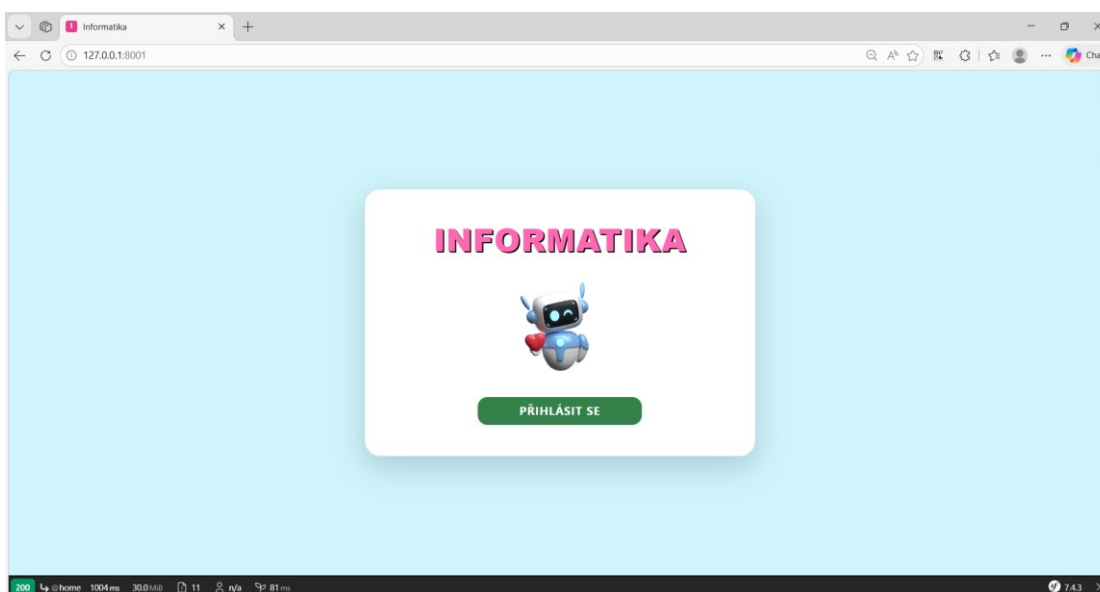
Při návrhu rozložení prvků byl kladen důraz na principy uživatelského designu, jejichž cílem je minimalizovat kognitivní úsilí žáka a usnadnit mu orientaci v aplikaci. Kognitivní zátěž vzniká zejména tehdy, pokud je uživatelské rozhraní nepřehledné, obsahuje nadbytečné prvky nebo neposkytuje jasné informace o tom, jakým způsobem má uživatel s aplikací pracovat. U dětí mladšího školního věku je proto nezbytné navrhovat rozhraní srozumitelně, s jasnou strukturou a předvídatelným chováním jednotlivých prvků.

Jak uvádí Norman (2010, s. 212), v rámci uživatelského designu je nezbytné zajistit, aby každý uživatel věděl:

- 1) co má udělat,
- 2) poznal, co přesně se v dané chvíli děje.

Zmíněné principy podporují pocit kontroly nad činností a snižují nejistotu uživatele při práci s digitálním prostředím. V předloženém návrhu jsou uvedené požadavky naplněny prostřednictvím jasné vizuální afordance akčního tlačítka, které svým vzhledem a umístěním naznačuje možnost interakce, a dále pomocí okamžité zpětné vazby systému po každé interakci žáka. Zpětná vazba informuje uživatele o tom, že jeho akce byla zaznamenána a správně provedena, tím je podpořen pocit jistoty, motivaci k další práci a přispívá k pozitivnímu uživatelskému prožitku.

Na obrázku 6 je znázorněna úvodní stránka aplikace, která slouží jako výchozí obrazovka pro práci žáka, učitele, koordinátora i administrátora.



Obr. 6: Úvodní stránka výukové aplikace

Zdroj: Vlastní zpracování

### 1.4.2 Psychologie barev

Použití barevné kombinace vycházelo z poznatků o psychologickém působení barev na lidské vnímání, emoce a schopnost soustředění. Barvy v uživatelském rozhraní neplní pouze estetickou funkci, ale slouží jako funkční nástroj podporující orientaci v aplikaci, udržení pozornosti a pozitivní emoční naladění žáků. Vhodně zvolená barevná paleta může přispět k lepší srozumitelnosti rozhraní a snížení kognitivní zátěže, což je zejména u mladších žáků klíčové.

Světle modré pozadí aplikace bylo zvoleno pro svůj uklidňující účinek, který přispívá k redukcí stresu a podporuje soustředění a kognitivní výkon žáků (Hrejsemnou, 2025, s. 19). Modrá barva je zároveň spojována s pocitem stability a bezpečí, napomáhá vytváření příjemného a předvídatelného uživatelského prostředí.

Motivační a akční prvky, například tlačítko „PŘIHLÁSIT“ a indikátory pokroku, jsou zvýrazněny zelenou barvou symbolizující správnost, bezpečí a postup v aktivitách. Zelená barva slouží jako vizuální signál úspěšného postupu a podporuje vnitřní motivaci žáků k dokončení zadaných úkolů (Hrejsemnou, 2025, s. 23).

Pro zvýšení hravosti a aktivizace žáků byly u vybraných navigačních prvků použity žluté a oranžové odstíny. Uvedené barvy jsou spojovány s optimismem, energií a pozitivním naladěním, čímž podporují ochotu žáků zapojit se do práce s obsahem a objevovat nové učivo (Hrejsemnou, 2025, s. 47).

Důležitou roli hraje také zachování dostatečného kontrastu mezi textem a pozadím a konzistentní používání barev v rámci celého rozhraní. Konzistence barev napomáhá rychlé orientaci v aplikaci, posiluje vizuální hierarchii prvků a přispívá k celkové přehlednosti uživatelského rozhraní.

## 2 Praktická část

Praktická část práce se zaměřuje na návrh, vývoj a testování webové aplikace. Popisuje strukturu systému, návrh databázového modelu, implementaci hlavních modulů a výsledky testování, které ověřují funkčnost aplikace. Součástí praktické části je také návrh možného budoucího rozšíření systému.

### 2.1 Struktura aplikace

Aplikace je rozdělena do několika vzájemně propojených částí, které umožňují žákům i učitelům pracovat s výukovým obsahem. Jednotlivé části systému zajišťují správu uživatelů, práci s výukovými lekci, vyplňování kvízů, evidenci docházky a sledování studijních výsledků.

Lekce a kvízy byly do systému vloženy pouze v omezeném rozsahu, především za účelem demonstrace a testování funkčnosti aplikace.

Ukázkové lekce byly vytvořeny s využitím nástroje NotebookLM. V reálném nasazení by výukový obsah připravovala odborně způsobilá osoba, například učitel nebo metodik informatiky.

Aplikace je navržena tak, aby jednotlivé moduly spolupracovaly, přičemž zůstávají přehledně oddělené.

Popis hlavních funkcí a rolí jednotlivých uživatelů je dále rozveden v následujících podkapitolách.

#### 2.1.1 Hlavní funkce

Aplikace je navržena jako komplexní nástroj, který propojuje tvorbu digitálního obsahu s jeho praktickým využitím v hodinách informatiky. Následující přehled shrnuje klíčové možnosti systému, které umožňují správu výuky, procvičování látky a efektivní sledování pokroku žáků.

##### **Seznam hlavních funkcí:**

- Personalizovaný přístup: Bezpečné přihlášení do systému a automatické zobrazení obsahu podle role uživatele a jeho ročníku.
- Interaktivní výuka: Práce s připravenými lekci a vyplňování kvízů.
- Administrace obsahu: Nástroje pro tvorbu a úpravu lekcí nebo interaktivních kvízů v editoru pro koordinátora.
- Sledování výsledků: Detailní přehled o úspěšnosti jednotlivců i tříd s možností exportu dat pro klasifikaci.
- Organizace výuky: Evidence docházky a přehled o důležitých termínech v kalendáři.

Základem systému jsou výukové lekce, které vytváří koordinátor na základě požadavků rámcového vzdělávacího programu (RVP). Připravený obsah je následně zpřístupněn ostatním uživatelům. Učitelé s vytvořenými materiály pracují přímo ve výuce, přičemž mají možnost doplňovat lekce o vlastní poznámky nebo rozšiřující informace podle aktuálních potřeb konkrétní třídy.

Důležitou součástí aplikace je mechanismus hodnocení a motivace. Žáci získávají přístup k lekcím, hrám a úkolům, které odpovídají jejich úrovni, a mohou tak pracovat vlastním tempem. Systém automaticky vyhodnocuje výsledky kvízů a zaznamenává pokrok, díky čemuž mají učitelé okamžitý přehled o tom, jak třída danou látku zvládá. Nasbíraná data lze navíc snadno exportovat pro další zpracování, čímž se výrazně zjednodušuje administrativa spojená s výukou.

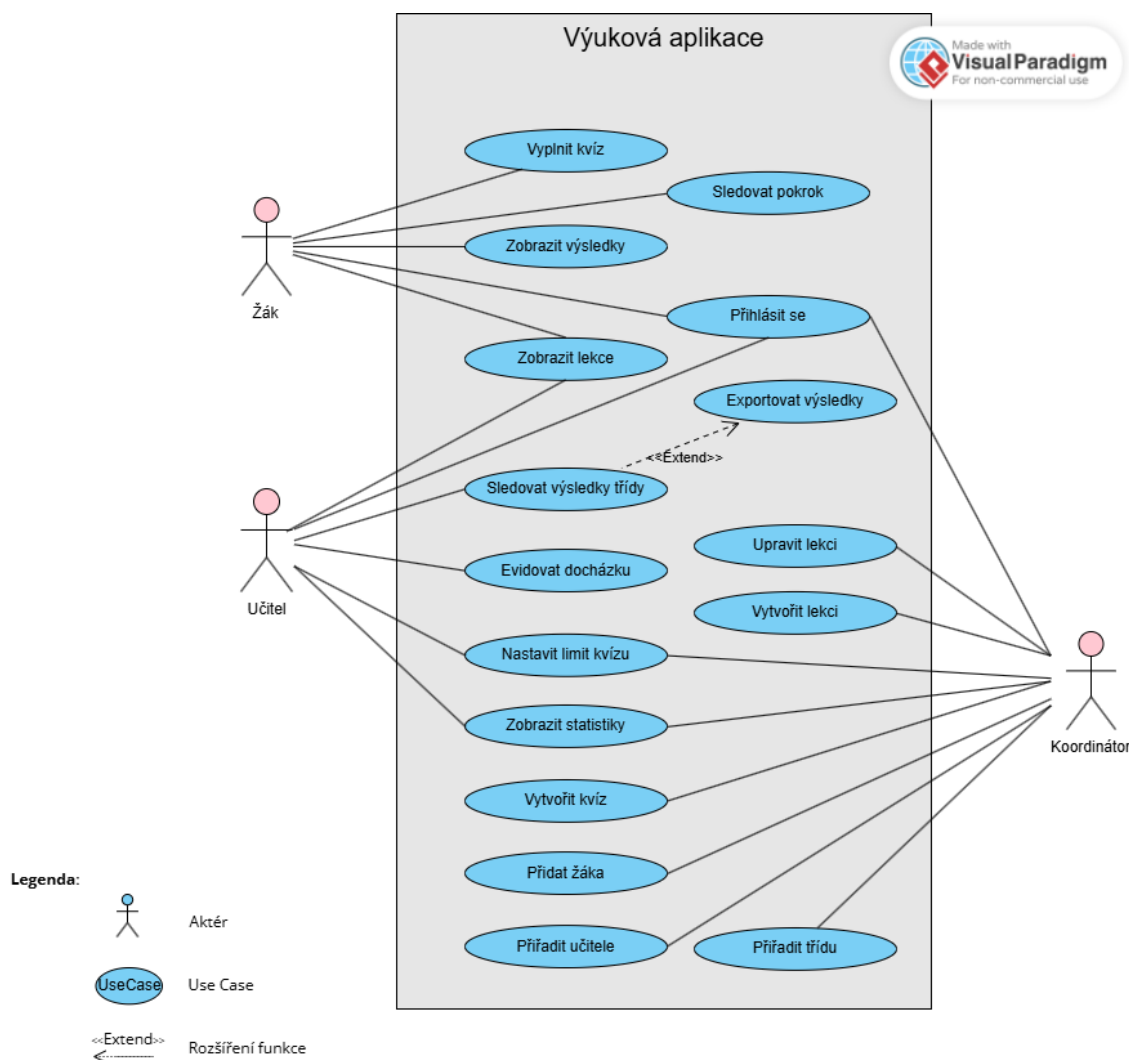
### 2.1.2 Role uživatelů

V aplikaci jsou definovány uživatelské role, které určují konkrétní úroveň oprávnění a rozsah dostupných činností pro jednotlivé skupiny uživatelů. Jedná se o hierarchický model, který zajišťuje vysokou úroveň bezpečnosti, přehlednou správu a garantuje, že každý uživatel má k dispozici nástroje nezbytné pro jeho specifické úkoly. Rozdělení rolí zároveň usnadňuje organizaci výukového procesu a efektivní distribuci obsahu mezi pedagogy a žáky.

#### **Systém rozlišuje čtyři základní uživatelské úrovně:**

- **Žák:** Představuje koncového uživatele výukového procesu. Má přístup k materiálům určeným pro jeho ročník, může procházet výukový obsah, plnit úkoly a využívat vzdělávací hry. Veškerý pokrok a výsledky kvízů jsou automaticky ukládány k jeho profilu pro pozdější vyhodnocení.
- **Učitel:** Působí jako mentor třídy, jehož hlavní funkcí je monitoring výsledků ve svěřených skupinách. Má oprávnění exportovat data o úspěšnosti, vést evidenci docházky a pracovat s připravenými materiály. Soustředí se primárně na distribuci obsahu a vyhodnocování pedagogických cílů.
- **Koordinátor:** Je klíčovou postavou pro tvorbu a správu obsahu. Má k dispozici nástroje pro vytváření nových lekcí a interaktivních kvízů v integrovaném builderu. Zajišťuje aktuálnost výukového programu a má pravomoc spravovat uživatelskou základnu.
- **Administrátor:** Představuje nejvyšší úroveň oprávnění s přístupem ke všem modulům systému. Zajišťuje technickou stabilitu, dohlíží na správnou funkci databáze a synchronizaci metadat.

Vzájemné interakce a toky dat mezi popisovanými rolemi tvoří uzavřený vzdělávací cyklus. Zatímco koordinátor obsah definuje, učitel jej spravuje a žák jej konzumuje. Dané vztahy zajišťují, že se každý uživatel může plně soustředit na svou roli v rámci digitálního vzdělávání.



Obr. 7: Příklad užití výukové aplikace

Zdroj: Vlastní zpracování

Diagram případů užití na obrázku 7 znázorňuje hlavní aktéry systému (žák, učitel, koordinátor) a jejich interakce s výukovou aplikací. Zároveň znázorňuje rozšíření některých funkcí, například možnost exportu výsledků.

## 2.2 Databáze

Databáze představuje stěžejní technologický prvek celé aplikace, jelikož zajišťuje bezpečné a strukturované ukládání všech dynamických dat. V systému jsou uchovávány informace o uživateli, jejich zařazení do tříd, kompletní výukový obsah, struktura kvízů i detailní záznamy o docházce a studijních výsledcích. Databázový model je navržen jako relační model umožňující efektivní propojování informací a minimalizaci duplicity dat.

Celé řešení je implementováno v systému MySQL. Komunikace mezi aplikační logikou v PHP a databázovým serverem probíhá skrze Doctrine ORM. Daný nástroj umožňuje vývojáři pracovat s databázovými záznamy jako s objekty, čímž se zrychluje vývoj, zvyšuje čitelnost kódu a je zajištěna automatická synchronizace metadat mezi entitami a fyzickými tabulkami v databázi.

### 2.2.1 Návrh databáze

Při tvorbě návrhu byl kladen důraz na logické oddělení jednotlivých modulů aplikace tak, aby systém zůstal přehledný i při budoucím rozšiřování o nové funkce. Základním pilířem návrhu jsou tabulky spravující identitu uživatelů a výukový obsah, na které jsou navázány moduly pro sledování pokroku a administrativu výuky.

Vztahy mezi entitami jsou realizovány pomocí relací typu One-To-Many (1: N) a Many-To-One (N:1) s využitím cizích klíčů, které garantují referenční integritu dat. To v praxi znamená, že systém nedovolí smazat záznam, na který odkazuje jiná část aplikace, čímž se předchází vzniku nekonzistentních dat.

#### Konkrétní ukázky propojení v aplikaci:

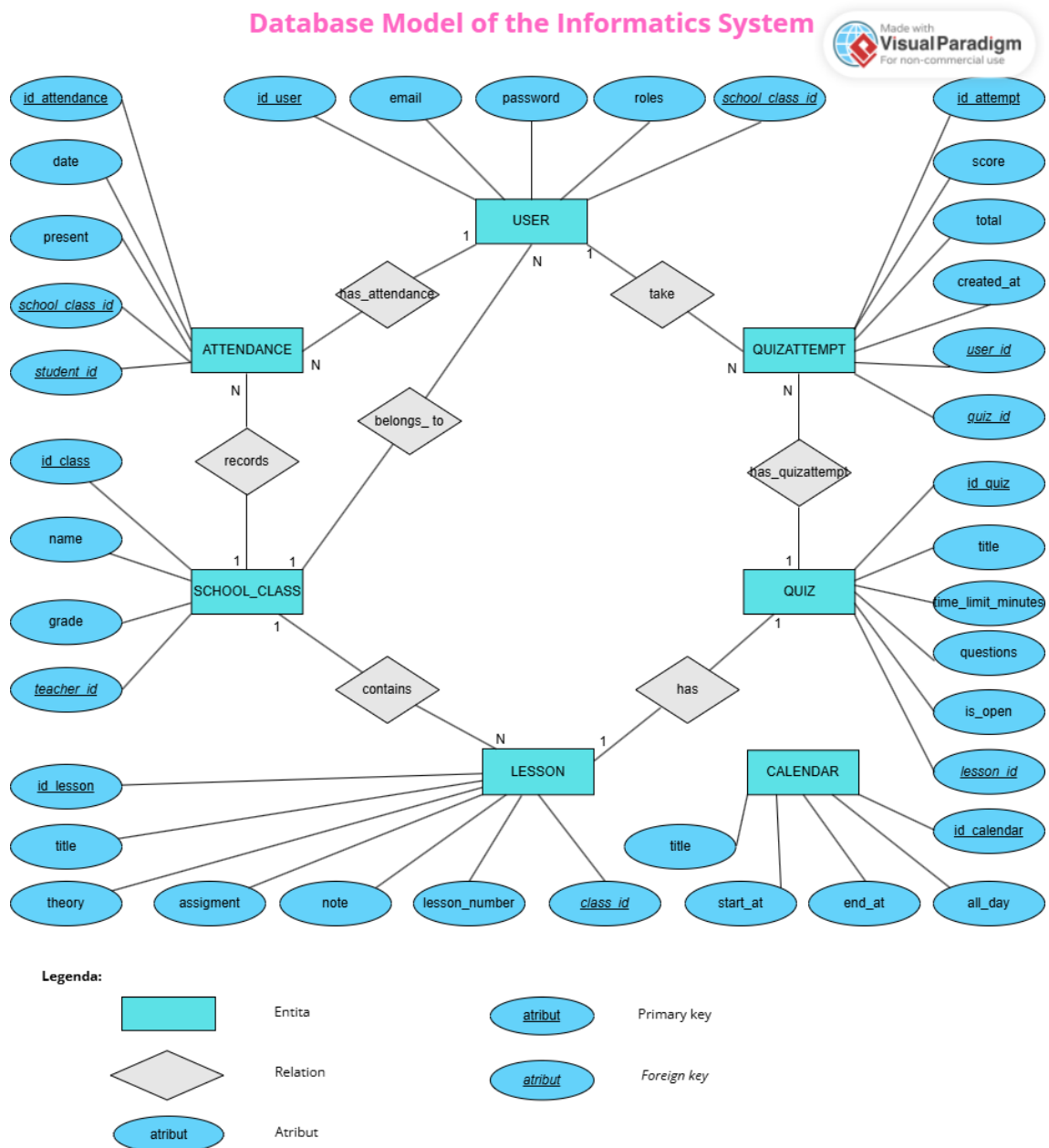
- Učitel a třídy: Jeden pedagog může spravovat více tříd najednou, zatímco každá třída má v systému jasně definovaného odpovědného učitele.
- Žáci a třídy: Každý žák je pevně přiřazen k jedné třídě, tím je umožněno automatické filtrování lekcí podle příslušného ročníku.
- Lekce a kvízy: Výukové lekce mohou být volitelně propojeny s interaktivními kvízy. Systém je navržen tak, aby u každého kvízu mohl žák provést více pokusů, přičemž každý pokus je samostatně evidován pro sledování dlouhodobého zlepšování znalostí.

**Tab. 2: Přehled hlavních databázových entit aplikace**

Entita	Popis
Users	Ukládá informace o uživatelích systému (žák, učitel, koordinátor).
SchoolClass	Obsahuje informace o třídách a jejich ročníku.
Lesson	Reprezentuje jednotlivé výukové lekce.
Quiz	Ukládá informace o kvízech.
QuizAttempt	Eviduje jednotlivé pokusy žáků o vyplnění kvízu.
Attendance	Slouží k evidenci docházky žáků na vyučovacích hodinách.
CalendarEvent	Ukládá události zobrazované v kalendáři aplikace.

*Zdroj: Vlastní zpracování*

Na základě identifikovaných entit a jejich základních atributů byl vytvořen ER model databáze, který graficky znázorňuje strukturu databázového návrhu a vztahy mezi jednotlivými entitami viz obrázek 8.



**Obr. 8: ER model**  
Zdroj: Vlastní tvorba

### 2.2.2 Identifikace entit a jejich klíčů

Následující část obsahuje podrobný rozbor jednotlivých entit databázového modelu, které tvoří logický základ aplikace. U každé položky je definován význam v rámci systému a uveden primární klíč (atribut id), který zajišťuje unikátnost záznamů a umožňuje vytváření relací mezi tabulkami.

USERS (Uživatel) Entita představuje ústřední prvek celého systému, neboť uchovává veškeré autentizační a autorizační údaje. Obsahuje atributy jako uživatelské jméno, heslo v bezpečné podobě a e-mailovou adresu. Zásadním parametrem je role, která určuje rozsah oprávnění uživatele – zda v systému vystupuje jako žák, učitel, koordinátor nebo administrátor. Primárním klíčem je atribut id.

SCHOOLCLASS (Školní třída) Reprezentuje organizační jednotku školy, tedy konkrétní třídu. Obsahuje název, a především informaci o ročníku. Uvedený atribut je stěžejní pro správné fungování aplikace, protože na jeho základě dochází k automatickému filtrování výukových lekcí a zadání úkolů určených pro danou skupinu žáků. Primárním klíčem je atribut id.

LESSON (Výuková lekce) Slouží k uložení kompletního výukového obsahu pro jednotlivé hodiny. Obsahuje název lekce, pořadové číslo hodiny v rámci tematického plánu a rozsáhlé pole pro samotný výukový obsah i zadání úkolů. Data jsou přímo mapována do uživatelského rozhraní žáka a učitele. Primárním klíčem je atribut id.

QUIZ (Kvíz) Definuje strukturu interaktivního ověřování znalostí. Kromě názvu obsahuje data o struktuře otázek a odpovědí, generovaná prostřednictvím vizuálního builderu. Každý kvíz je v aplikaci přiřazen ke konkrétní lekci prostřednictvím relační vazby, která umožňuje zobrazovat žákům pouze kvízy odpovídající jejich ročníku. Důležitým parametrem je nastavení časového limitu, omezující dobu trvání pokusu. Primárním klíčem je atribut id.

QUIZATTEMPT (Pokus o vyplnění kvízu) Představuje hlavní nástroj pro sledování úspěšnosti vzdělávacího procesu. Zaznamenává každý jednotlivý pokus žáka o vypracování kvízu, přičemž ukládá dosažené skóre, celkový počet možných bodů a přesný časový údaj o odevzdání. Uvedená data umožňují učitelům analyzovat pokrok žáků v čase. Primárním klíčem je atribut id.

ATTENDANCE (Docházka) Entita určená pro administrativní správu vyučovacích hodin. Eviduje přítomnost či nepřítomnost konkrétního žáka v určeném čase a hodině. Úzké propojení s tabulkou uživatelů umožňuje generování statistik docházky pro jednotlivce i celé třídy. Primárním klíčem je atribut id.

CALENDAR (Kalendář událostí) Reprezentuje plánované události zobrazované v rozhraní aplikace. Obsahuje název události (např. termín odevzdání úkolu nebo školní akce). Slouží k lepší organizaci času a informovanosti učitelů i žáků. Primárním klíčem je atribut id.

### 2.2.3 Identifikace vztahů

Klíčovou součástí návrhu je definice vztahů mezi entitami, vyjadřujících logické propojení objektů a způsob vzájemného ovlivňování dat. Relace jsou implementovány prostřednictvím cizích klíčů, které určují kardinalitu mezi tabulkami a zajišťují integritu informací napříč celým systémem.

#### Popis konkrétních relací v systému:

„Patří\_do“ (USERS a SCHOOLCLASS): Definuje přiřazení žáka ke konkrétní třídě. Žák náleží nejvýše k jedné třídě, zatímco třída sdružuje více žáků. Z pohledu třídy se jedná o kardinalitu 1: N.

„Vyučuje“ (USERS a SCHOOLCLASS): Určuje vazbu mezi učitelem a třídou. Učitel spravuje více tříd, každá třída má však přiděleného právě jednoho pedagoga. Kardinalita vztahu je 1: N.

„Obsahuje“ (SCHOOLCLASS a LESSON): Vyjadřuje vazbu výukových lekcí k ročníkům. Třída disponuje přístupem k většímu počtu lekcí, každá lekce je však unikátně přiřazena jedné třídě. Kardinalita odpovídá poměru 1: N.

„Má\_kvíz“ (LESSON a QUIZ): Propojuje konkrétní lekci s testovým modulem. Jedna lekce obsahuje maximálně jeden kvíz a kvíz náleží právě jedné lekci. Kardinalita je definována jako 1: 1, přičemž vytvoření kvízu pro lekci zůstává volitelné.

„Vyplní“ (USER a QUIZATTEMPT): Zachycuje historii interakce žáka s testy. Žák generuje více pokusů, každý pokus je však svázán s identitou právě jednoho žáka. Kardinalita vztahu je 1: N.

„Náleží\_ke\_kvízu“ (QUIZ a QUIZATTEMPT): Udává příslušnost pokusu k šabloně kvízu. Jeden kvíz uchovává historii mnoha pokusů o vyplnění. Kardinalita dosahuje poměru 1: N.

„Má\_docházku“ (USER a ATTENDANCE): Slouží k evidenci přítomnosti žáka. Žák disponuje historií více záznamů docházky, každý záznam je však unikátně přiřazen konkrétní osobě. Kardinalita vztahu je 1: N.

### Hlavní typy dotazů v systému

Aplikace využívá databázové dotazy pro získávání a zpracování informací potřebných pro běžný provoz systému. Mezi nejdůležitější operace patří zejména:

- zobrazení výukových lekcí podle ročníku nebo třídy přihlášeného žáka,
- zobrazení kvízů přiřazených k jednotlivým lekcím přihlášeného žáka,
- výpis seznamu žáků přiřazených k jednotlivým třídám pro potřeby učitelů,
- vyhledání historie pokusů o vyplnění konkrétního kvízu,
- zobrazení výsledků žáků a jejich úspěšnosti v jednotlivých kvízech,
- výpočet statistik docházky pro jednotlivce i celé třídní kolektivy,
- filtrování událostí v kalendáři podle aktuálního data,
- export dat s výsledky kvízů do souboru formátu CSV.

### 2.2.4 Relační model databáze

Relační model databáze vychází z předchozího ER modelu a představuje konkrétní implementaci databázové struktury ve formě entit (tabulek). Každá entita z ER modelu je v relačním modelu reprezentována samostatnou databázovou tabulkou, která obsahuje jednotlivé atributy a primární klíč.

Vztahy mezi entitami jsou realizovány pomocí cizích klíčů, které propojují jednotlivé tabulky databáze. Například tabulka *users* obsahuje odkaz na tabulku *schoolclass*, který určuje, do jaké třídy je uživatel přiřazen. Tabulka *lesson* je propojena s tabulkou *schoolclass*, čímž je umožněno přiřazení výukové lekce ke konkrétním třídám.



Aplikace je implementována v programovacím jazyce PHP a k ukládání dat využívá relační databázi MySQL. Uživatelské rozhraní kombinuje jazyk HTML, kaskádové styly CSS (s využitím frameworku Bootstrap pro responzivitu) a pro pokročilé interaktivní prvky, jako je například builder kvízů, také jazyk JavaScript. Při návrhu byl kladen důraz na modularitu a logické oddělení funkčních částí systému.

**Mezi hlavní moduly aplikace patří zejména:**

- modul správy uživatelů
- modul výukových lekcí
- modul kvízů a vyhodnocování výsledků
- modul evidence docházky
- modul kalendáře

Modul výukových lekcí umožňuje koordinátorovi vytvářet a upravovat výukový obsah, který je následně dostupný učitelům a žákům. Modul kvízů slouží k ověřování znalostí žáků a zaznamenává jejich výsledky do databáze. Učitelé mohou sledovat výsledky své třídy a v případě potřeby je exportovat. Součástí aplikace je také modul evidence docházky, který umožňuje učitelům zaznamenávat přítomnost žáků na jednotlivých hodinách. Další část systému tvoří kalendář, ve kterém lze zobrazovat důležité události související s výukou.

Součástí aplikace je také modul pro evidenci docházky, který umožňuje učitelům zaznamenávat přítomnost žáků na jednotlivých hodinách. Další částí systému je kalendář, ve kterém lze zobrazovat důležité události související s výukou.

### 2.3.1 Dashboard podle role

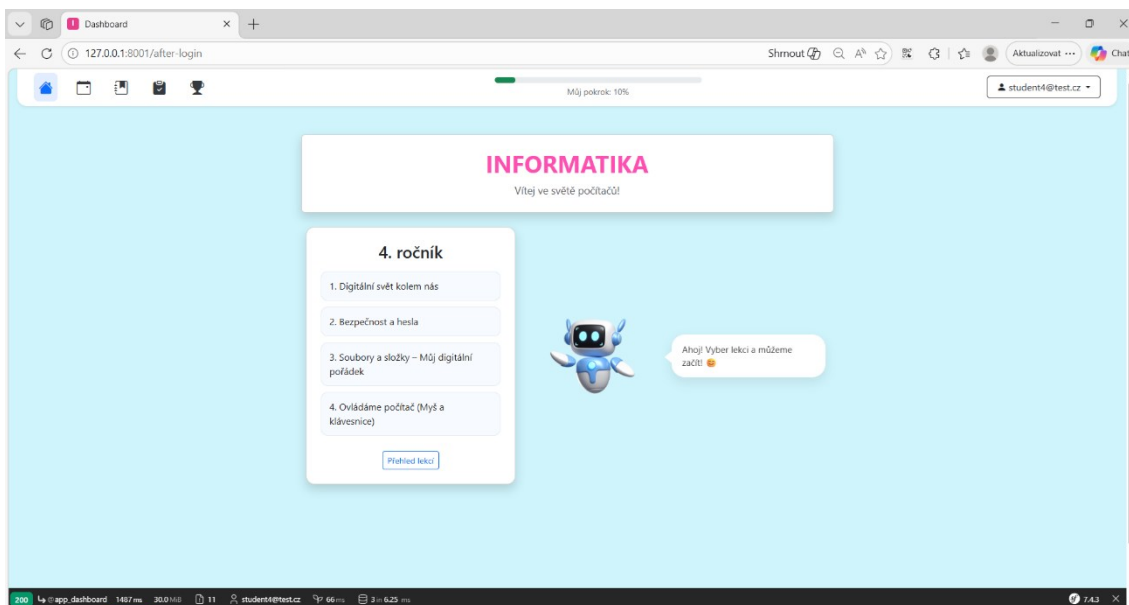
Po úspěšné autentizaci dochází k automatickému přesměrování uživatele na hlavní stránku, jejíž obsah se dynamicky přizpůsobuje přidělené roli. Dashboard představuje výchozí komunikační rozhraní a zajišťuje okamžitý přístup k nejčastěji využívaným funkcím uživatele.

**Vizuální i funkční podoba hlavní stránky reflektuje specifické potřeby jednotlivých skupin:**

- **Žáci:** Rozhraní se zaměřuje na snadnou dostupnost lekcí a kvízů odpovídajících příslušnému ročníku. Ústředním prvkem je vizuální ukazatel pokroku, který žáka informuje o splněných úkolech a úspěšně absolvovaných testech. Hlavní strana umožňuje přímý vstup do aktuální výuky bez nutnosti využívat složité navigační struktury.
- **Učitelé:** Hlavní přehled nabízí souhrnné informace o svěřené třídě, aktuální docházce a studijních výsledcích. Rozhraní slouží jako odrazový můstek pro administrativní úkony, jako je evidence přítomnosti, detailní analýza úspěšnosti jednotlivců nebo export dat třídy pro potřeby klasifikace.
- **Koordinátor:** Disponuje nástroji pro komplexní správu vzdělávacího obsahu. Rozhraní umožňuje rychlý přechod do editoru lekcí, vizuálního builderu kvízů a sekce pro správu uživatelských účtů včetně přiřazování pedagogů ke třídám.

Celkové pojetí UI klade důraz na intuitivnost a přehlednost. Cílem je minimalizovat kognitivní zátěž, aby byla aplikace snadno ovladatelná i pro méně zkušené uživatele. Navigace využívá kombinaci srozumitelných ikon a horizontálního či vertikálního menu, zrychluje se tak interakce s jednotlivými moduly.

Ukázky dashboardů pro jednotlivé role uživatelů jsou zobrazeny na následujících obrázcích.

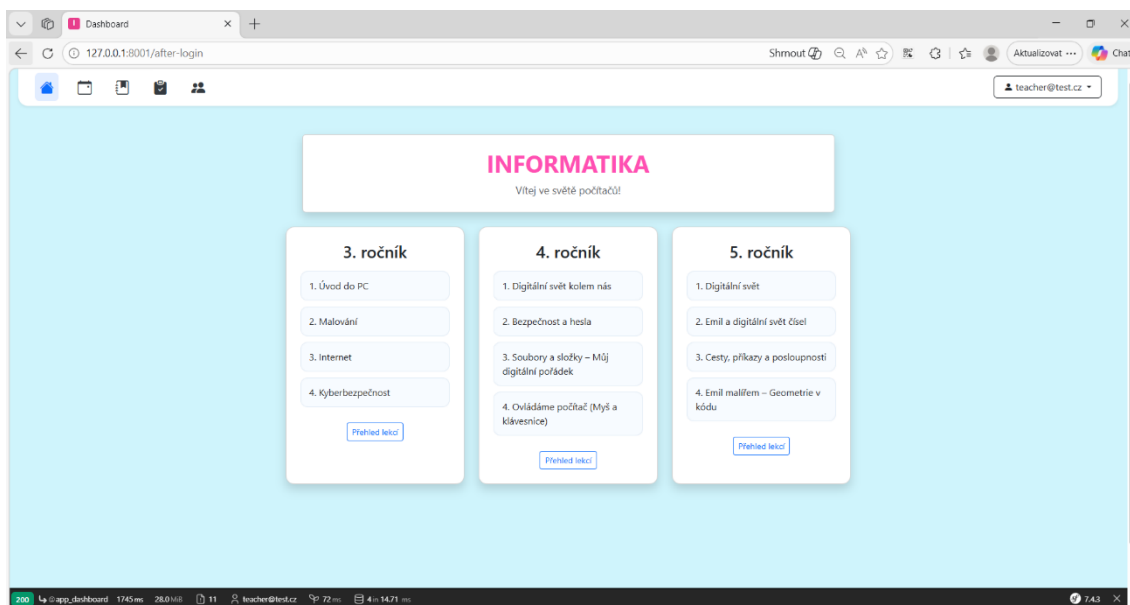


**Obr. 10: Dashboard žáka**

*Zdroj: Vlastní zpracování*

UI určené pro žáky (Obrázek 10) klade důraz na maximální přehlednost a intuitivní ovládání, zohledňující specifické potřeby mladších uživatelů. Dominantním prvkem hlavní strany je karta s přehledem lekcí pro příslušný ročník, která umožňuje rychlé zahájení výuky bez nutnosti hloubkového procházení navigačního menu. Prostředí je doplněno o grafický prvek robota s interaktivní bublinou, který uživatele vítá a navigací ho provází, čímž snižuje kognitivní zátěž.

V horní části rozhraní je umístěn vizuální ukazatel pokroku (progress bar), který plní motivační funkci. Žákovi poskytuje okamžitou zpětnou vazbu o procentuálním postupu v rámci probírané látky a orientačně znázorňuje průběh vzdělávání. Horní lišta dále obsahuje ikony pro rychlý přístup ke kalendáři, lekcím, kvízům a výsledkům, čímž je podpořena samostatnost žáků při práci s aplikací. Celkové barevné ladění a uspořádání prvků podporuje přehlednost a snadnou orientaci v systému.

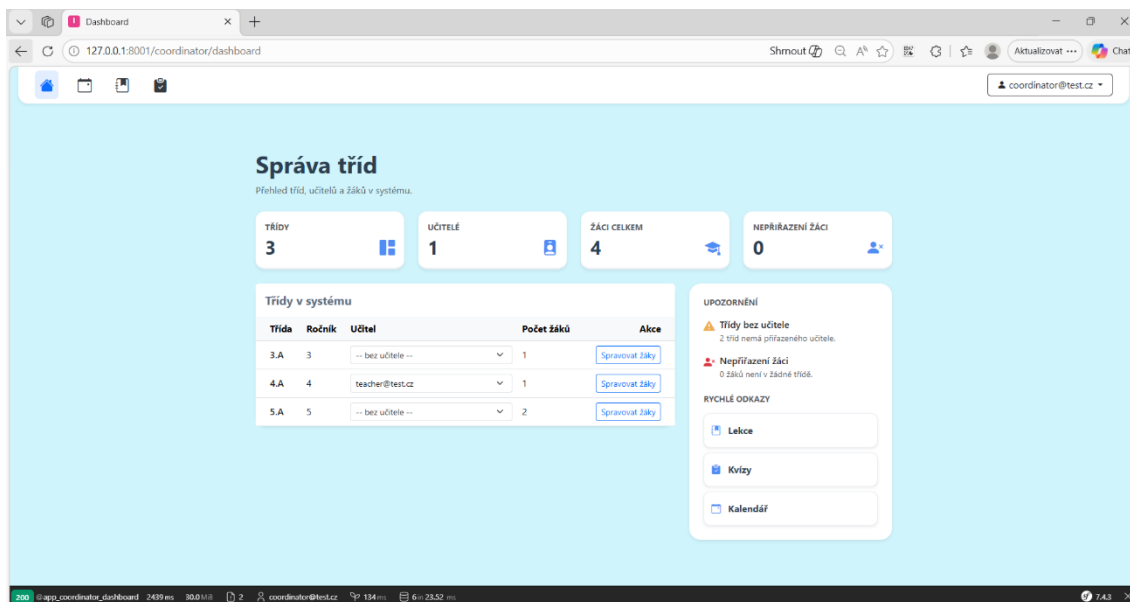


**Obr. 11: Dashboard učitele**

*Zdroj: Vlastní zpracování*

Obrázek 11 zobrazuje UI pro učitele. Poskytuje ucelený přehled výukových lekcí strukturovaných podle jednotlivých ročníků. Na rozdíl od žákovského pohledu disponuje pedagog přístupem ke všem dostupným materiálům napříč celým vzdělávacím programem.

Horní navigační lišta obsahuje identické ovládací prvky jako u žákovské role, avšak s rozšířenými ikonami pro správu tříd a sledování statistik úspěšnosti. Celkové grafické pojetí zachovává vizuální identitu aplikace, čímž zajišťuje konzistentní uživatelskou zkušenost napříč všemi úrovněmi systému.



**Obr. 12: Dashboard koordinátora**

*Zdroj: Vlastní zpracování*

Dashboard koordinátora slouží jako centrální rozhraní pro správu výukového systému a sledování statistik o počtech tříd, pedagogů i žáků.

V prostřední části obrazovky se nachází seznam všech tříd, které jsou v systému uloženy. U každé třídy lze vidět její ročník, jméno učitele a aktuální počet žáků. Tlačítko „Spravovat žáky“ slouží k otevření detailu, kde koordinátor přidává nové žáky nebo je rozděluje do jednotlivých tříd.

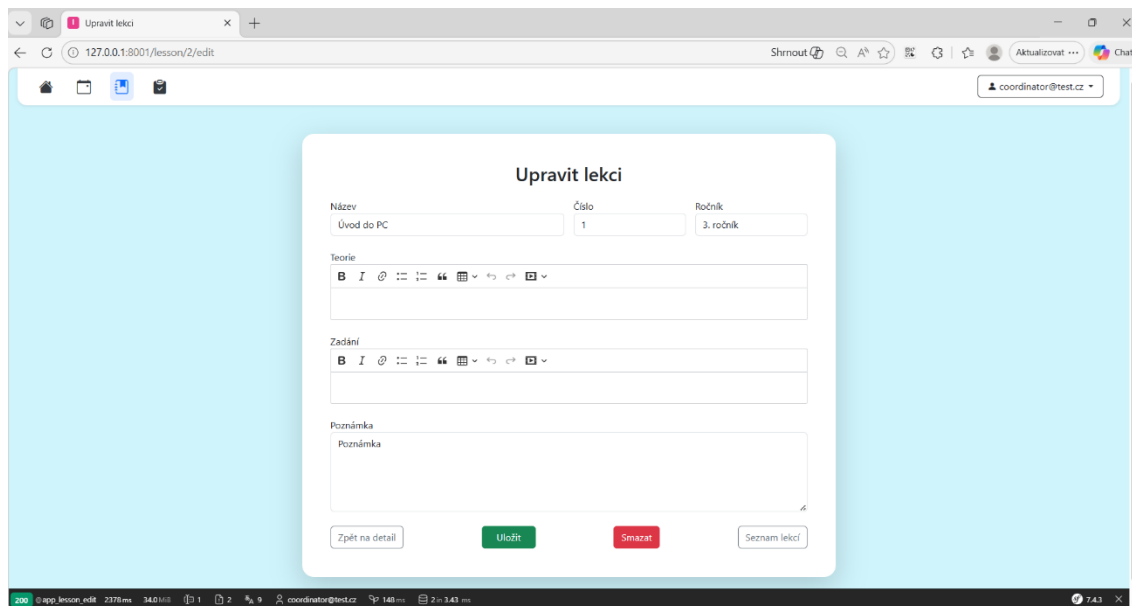
Pravá strana plochy nabízí panel s rychlými odkazy, které urychlují pohyb v aplikaci. Je to část usnadňující přechod k úpravám lekcí, tvorbě kvízů nebo k prohlížení kalendáře, šetří čas při běžné správě webu.

### 2.3.2 Editor úloh

Součástí aplikace je editor úloh určený pro vytváření výukových lekcí v souladu se školním vzdělávacím programem (ŠVP) dané školy. Nástroj využívá především koordinátor výuky informatiky, který připravuje strukturu lekcí a zajišťuje, aby obsah odpovídal požadavkům rámcového vzdělávacího programu (RVP). Učitelé mají možnost k jednotlivým lekcím přidávat vlastní poznámky, které slouží k doplnění metodických informací, rozšíření obsahu hodiny nebo upřesnění vzdělávacích cílů.

Na obrázku 13 je zobrazen editor pro vytvoření nové lekce. Rozhraní umožňuje zadat název lekce, číslo hodiny a příslušný ročník. Formulář dále obsahuje pole pro vložení teoretické části výuky a praktického zadání pro žáky.

Pro úpravu textu slouží WYSIWYG editor CKEditor, který umožňuje formátování textu, práci se seznamy, vkládání odkazů a multimediálního obsahu. Do lekcí lze vkládat také videozáznamy, které slouží jako doplňující vysvětlení učiva nebo praktická ukázka práce s digitálními nástroji.



**Obr. 13: Editor lekcí**  
Zdroj: Vlastní zpracování

### Technická implementace vkládání multimediálního obsahu

V rámci editoru úloh byla implementována funkce pro automatické zpracování vložených video odkazů. Příložená ukázka kódu na obrázku 14 v jazyce JavaScript demonstruje logiku extrakce unikátního identifikátoru videa (ID) z adresy URL. Skript nejprve ověřuje přítomnost domény youtube.com neboyoutu.be v zadaném řetězci.

#### Následně algoritmus provádí parsování adresy:

- Pokud odkaz obsahuje parametr "v", systém získá ID videa z řetězce za tímto oddělovačem.
- V případě zkrácených odkazů skript izoluje identifikátor jako poslední část cesty za lomítkem.

Získané ID slouží k vygenerování responzivního přehrávače přímo v obsahu lekce, což učitelům umožňuje snadno obohatit výuku o doplňující videozáznamy bez nutnosti manuálního nastavování vkládacích kódů. Uvedené řešení zvyšuje uživatelský komfort koordinátora při přípravě materiálů a zajišťuje vizuální stabilitu prezentovaného obsahu pro žáky.

```
embeds.forEach(embed => {
  const url = embed.getAttribute("url");
  if (!url) return;

  if (url.includes("youtube.com") || url.includes("youtu.be")) {
    let videoId = "";

    if (url.includes("v=")) {
      videoId = url.split("v=")[1] || "";
    } else {
      videoId = url.split("/").pop() || "";
    }
  }
});
```

Obr. 14: Ukázka zpracování odkazu na video v JavaScriptu

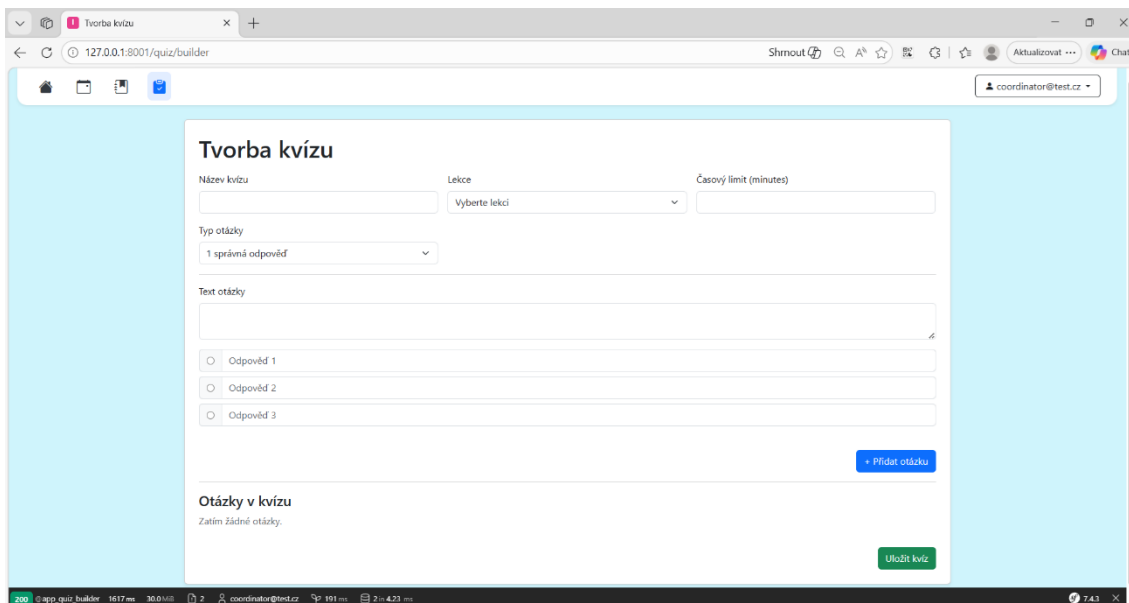
*Zdroj: Vlastní zpracování*

### 2.3.3 Editor Kvízů

Součástí systému je specializovaný editor pro tvorbu interaktivních kvízů, které slouží k ověřování znalostí žáků. Nástroj využívá primárně koordinátor pro přípravu obsahu a definování testových otázek. Rozhraní umožňuje zadat název kvízu, stanovit časový limit pro vypracování a postupně přidávat jednotlivé otázky s jednou nebo více variantami odpovědí, u kterých se označuje správná volba.

Data jsou během tvorby ukládána do dynamické struktury v klientské části aplikace a následně odesílána na server ve formátu JSON. Významnou funkcí je možnost úpravy časového limitu učitelem, což dovoluje přizpůsobit náročnost testu potřebám konkrétní třídy a podporuje individuální přístup k výuce.

Dostupnost kvízu pro žáky řídí učitel skrze parametry otevření nebo uzavření testu, čímž je zajištěno, že se žáci k obsahu nedostanou mimo plánovaný čas vyučování. Prostředí editoru zachycuje obrázek 15.



**Obr. 15: Editor kvízů**

*Zdroj: Vlastní zpracování*

### Implementace ukládání dat

Po odeslání dat z klientské části dochází v kontroloru aplikace k dekodování JSON struktury a k validaci vstupů. Server ověřuje zejména přítomnost názvu a existenci alespoň jedné otázky, v případě nesrovnalostí vrací chybovou odpověď.

Při úspěšné validaci systém vytvoří novou instanci entity *Quiz*, do které uloží název a definovanou strukturu otázek. Jak demonstruje ukázka zdrojového kódu na obrázku 16, po vytvoření objektu dochází k nastavení základních informací.

Nově vytvořený záznam je standardně nastaven jako uzavřený (`$quiz->setIsOpen(false)`), což ponechává rozhodnutí o zpřístupnění na vyučujícím. Finální uložení do databáze MySQL probíhá prostřednictvím nástroje Doctrine ORM, kde jsou změny potvrzeny metodou `flush()`.

```
$quiz = new Quiz();
$form = $this->createForm(QuizType::class, $quiz);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    $entityManager->persist($quiz);
    $entityManager->flush();

    return $this->redirectToRoute('app_quiz_index', [], Response::HTTP_SEE_OTHER);
}
```

**Obr. 16: Ukázka ukládání kvízu do databáze v kontroloru aplikace**

*Zdroj: Vlastní zpracování*

### 2.3.4 Evidence docházky žáků

Dalším funkčním celkem aplikace je modul evidence docházky, který učitelům umožňuje zaznamenávat přítomnost žáků v rámci jednotlivých vyučovacích hodin. Docházka je evidována vždy pro konkrétní třídu, ke které má vyučující přiděleno oprávnění.

Rozhraní nabízí přehledný seznam žáků s interaktivním formulářem pro označení přítomnosti. Jednotlivé záznamy jsou prezentovány formou zaškrťovacích políček (checkboxů), která jsou ve výchozím stavu nastavena jako aktivní. Uvedené nastavení dovoluje učiteli provádět rychlé úpravy pouze u žáků, kteří ve výuce reálně chybí. Pro zvýšení efektivity práce obsahuje formulář funkci realizovanou v jazyce JavaScript, která umožňuje hromadné označení všech studentů jako přítomných.

```

{% for s in students %}
  {% if 'ROLE_STUDENT' in s.roles %}
    <div class="form-check mb-2">
      <input class="form-check-input"
        type="checkbox"
        name="students[]"
        value="{{ s.id }}"
        id="student_{{ s.id }}"
        checked>
      <label class="form-check-label ms-2"
        for="student_{{ s.id }}">
        {{ s.email }}
      </label>
    </div>
  {% endif %}
{% endfor %}

```

Obr. 17: Ukázka generování seznamu žáků ve formuláři docházky

Zdroj: Vlastní zpracování

#### Technická implementace formuláře

Zpracování seznamu žáků probíhá dynamicky na straně serveru s využitím šablonovacího systému Twig. Jak dokumentuje ukázka zdrojového kódu na obrázku 17, generování prvků formuláře zajišťuje cyklus `{% for %}`, který prochází pole uživatelů přiřazených k dané třídě. Pro zajištění datové správnosti je implementována podmínka `{% if 'ROLE_STUDENT' in s.roles %}`, která filtruje výhradně uživatele s rolí žáka. Pro každou nalezenou entitu je následně vytvořeno pole s unikátním identifikátorem žáka (`s.id`) a jeho e-mailovou adresou (`s.email`).

Po odeslání dat serverová část aplikace zpracuje seznam identifikátorů přítomných žáků a vytvoří v databázi nový záznam docházky, obsahující vazbu na třídu a konkrétní datum výuky. Mechanismus dovoluje následnou práci se získanými daty při generování statistik nebo přehledů.

#### Statistické přehledy účasti

Součástí modulu je také sekce statistických přehledů, která pedagogům poskytuje souhrnné informace o docházce celých třídních kolektivů i jednotlivců. V horní části rozhraní jsou integrovány základní ukazatele, jako je celkový počet žáků ve třídě, počet evidovaných vyučovacích hodin a průměrná procentuální účast. Uvedená data slouží k rychlé diagnostice absencí a usnadňují administrativu spojenou s vedením pedagogické dokumentace.

```

{% set studentsCount = stats|length %}
{% set totalLessons = 0 %}
{% set totalPresent = 0 %}
{% set totalAll = 0 %}

{% for row in stats %}
  {% set totalPresent = totalPresent + (row.presentCount ?? 0) %}
  {% set totalAll = totalAll + (row.total ?? 0) %}
  {% if (row.total ?? 0) > totalLessons %}
    {% set totalLessons = row.total %}
  {% endif %}
{% endfor %}

```

**Obr. 18: Ukázka výpočtu souhrnných statistik docházky v šabloně Twig**

*Zdroj: Vlastní zpracování*

Modul docházky obsahuje algoritmus pro automatický výpočet souhrnných statistik, který je realizován přímo v šablonovacím systému Twig. Výpočet probíhá agregací dat o jednotlivých žácích, ze kterých systém postupně načítá hodnoty o celkovém počtu evidovaných hodin a počtu reálných přítomností.

Jak dokumentuje ukázka zdrojového kódu na obrázku 18, šablona nejprve inicializuje pomocné proměnné pro uložení celkových součtů. Následně cyklus `{% for row in stats %}` prochází jednotlivé záznamy a provádí kumulativní součet přítomností (`totalPresent`) a všech evidovaných termínů (`totalAll`). Pro ošetření prázdných hodnot je využit operátor nullable (`null coalescing`), který zajišťuje stabilitu výpočtu i v případě chybějících dat. Logickou správnost celého modulu pak doplňuje filtrování uživatelů pomocí podmínky `{% if 'ROLE_STUDENT' in s.roles %}`, která zaručuje, že statistiky zahrnují výhradně žáky, a nikoliv pedagogický dozor.

Konečná fáze výpočtu určuje průměrnou procentuální účast třídy (`avgAttendance`). Logika zahrnuje kontrolu nulového dělitele, po které dochází k podílu přítomností vůči celkovému počtu hodin a následnému vynásobení stem. Výsledné hodnoty jsou využity pro zobrazení statistických ukazatelů v horní části uživatelského rozhraní.

Uvedený přehled umožňuje učitelům okamžitě identifikovat průměrnou docházku celého kolektivu i jednotlivců, což usnadňuje včasné rozpoznání problémů s pravidelnou účastí žáků ve výuce.

### 2.3.5 Ukazatel pokroku

Ukazatel pokroku představuje vizuální prvek UI, který informuje žáka o jeho aktivitě v rámci výukového systému. Prvek plní především orientační a motivační funkci, neboť žákovi poskytuje okamžitou zpětnou vazbu o jeho interakci s aplikací. V aktuální implementaci je indikátor založen na logickém příznaku, který ověřuje, zda žák již odevzdal alespoň jeden kvíz.

Zobrazení ukazatele je striktně vázáno na roli uživatele a je dostupné výhradně žákům. Ostatní role, jako je učitel, koordinátor nebo administrátor, nejsou tímto prvkem zatěžovány, čímž dochází k personalizaci rozhraní podle reálných potřeb jednotlivých typů rolí. Zvolené řešení zajišťuje, že UI zůstává přehledné, což výrazně zvyšuje celkové UX systému, protože každý uživatel vidí pouze informace relevantní pro svou roli.

Z technického hlediska je prvek realizován pomocí komponenty progress bar z frameworku Bootstrap. Jak dokumentuje ukázka zdrojového kódu na obrázku 19, řízení přístupu probíhá přímo v šabloně Twig pomocí bezpečnostní funkce `is_granted('ROLE_STUDENT')`. Kód využívá standardní třídy Bootstrapu pro zajištění konzistentního vzhledu a responzivity na různých typech zařízení. Na základě hodnoty v proměnné progress se dynamicky nastavuje šířka výplně.

```
{% set progress = progress ?? (app.user.hasSubmittedQuiz ? 10 : 0) %}

<div class="d-flex flex-column align-items-center">
  <div class="progress mb-1" style="height: 10px; width: 350px; max-width: 40vw;">
    <div class="progress-bar bg-success"
      role="progressbar"
      style="width: {{ progress }}%;"
      aria-valuenow="{{ progress }}"
      aria-valuemin="0"
      aria-valuemax="100">
    </div>
  </div>

  <small class="text-muted">
    Můj pokrok: {{ progress }}%
    {% if progress == 100 %}  {% endif %}
  </small>
</div>
```

**Obr. 19: Ukázka implementace ukazatele pokroku v šabloně Twig**

*Zdroj: Vlastní zpracování*

### 2.3.6 Hodnocení žáků a export výsledků

Součástí aplikace je modul pro sledování a vyhodnocování výsledků kvízů. Po vyplnění testu dochází k automatickému uložení záznamu do databáze. Záznam obsahuje dosažené skóre, celkový počet bodů a přesné datum odevzdání. Zvolený postup umožňuje uchovávat historii pokusů a následně ji prezentovat uživatelům systému v přehledné formě.

Žák má k dispozici samostatnou stránku s přehledem vlastních výsledků, kde může sledovat úspěšnost u jednotlivých úloh. Rozhraní zobrazuje název kvízu, počet bodů, procentuální vyjádření a časové razítko odevzdání. Popsaná část UI dovoluje žákovi monitorovat vlastní postup v čase. Každá interakce s kvízem tvoří v databázi unikátní záznam, který zajišťuje integritu historických dat.

Učiteli je zpřístupněn souhrnný přehled výsledků všech žáků pro konkrétní kvíz. Pod náhledem kvízu, který zobrazuje základní informace a seznam otázek, je umístěno modré tlačítko „Výsledky žáků“. Prvek funguje jako přímý navigační uzel vedoucí na detailní tabulku úspěšnosti celého kolektivu.

Samotná stránka výsledků nabízí tabulku s identifikací žáka, posledním i nejlepším dosaženým skóre a celkovým počtem pokusů. Jak dokumentuje ukázka zdrojového kódu na obrázku 20, k vykreslení dat dochází dynamicky v šabloně Twig prostřednictvím cyklu, který prochází předané pole záznamů. Přímo pod hlavní tabulkou jsou umístěny ovládací prvky, včetně tlačítka pro návrat a zeleného tlačítka „Exportovat (CSV)“.

```
{% if rows is empty %}
<p class="text-muted mb-0">Zatím žádné výsledky.</p>
{% else %}
<div class="table-responsive">
<table class="table align-middle mb-0">
<thead>
<tr>
<th>Žák</th>
<th>Poslední</th>
<th>Nejlepší</th>
<th>Pokusy</th>
<th>Naposledy</th>
</tr>
</thead>
<tbody>
{% for r in rows %}
<tr>
<td class="fw-semibold">{{ r.user.email }}</td>
<td>{{ r.lastScore }}/{{ r.lastTotal }}</td>
<td>{{ r.bestScore }}/{{ r.bestTotal }}</td>
<td>{{ r.attemptCount }}</td>
<td>{{ r.lastAt|date('d.m.Y H:i') }}</td>
</tr>
{% endfor %}
</tbody>
</table>
```

**Obr. 20: Ukázka generování tabulky výsledků žáků v šabloně Twig**

*Zdroj: Vlastní zpracování*

Zatímco UI přehledu v aplikaci zobrazuje komplexní statistiky včetně historie všech pokusů, exportní skript je navržen tak, aby pro každého žáka generoval pouze jeho nejlepší dosažený výsledek. Implementovaná logika výrazně zvyšuje UX celého systému, protože pedagog získá čistý soubor připravený pro přímou klasifikaci bez nutnosti manuální filtrace dat v tabulkovém procesoru.

### 2.3.7 Kalendář

Kalendář slouží k přehledu a správě klíčových událostí souvisejících s výukou, jako jsou plánované úkoly, termíny nebo organizační sdělení. Modul rozšiřuje aplikaci o plánovací nástroj, který zvyšuje přehlednost práce v systému a přispívá k lepší orientaci uživatele. Implementace využívá JavaScriptovou knihovnu FullCalendar, která umožňuje zobrazení událostí v měsíčním přehledu s lokalizací do českého jazyka a nastavením týdne začínajícího pondělím.

Data jsou do rozhraní načítána dynamicky z backendu, takže zobrazované informace zůstávají aktuální. Uživatel může vytvořit záznam kliknutím na vybrané datum, čímž dojde k vyvolání dialogu pro zadání názvu. Po potvrzení jsou údaje odeslány na server, přičemž úspěšné uložení vede k okamžité aktualizaci UI bez nutnosti znovunačtení stránky. Zvolený postup zajišťuje plynulé UX a intuitivní ovládání celého modulu.

Technické pozadí komunikace mezi prohlížečem a serverem dokumentuje ukázka zdrojového kódu na obrázku 21. Pro přenos dat je využito rozhraní fetch API, které odesílá požadavky s tělem ve formátu JSON. Informace o názvu jsou serializovány pomocí funkce JSON.stringify() a následně zpracovány serverovou částí aplikace. Integrace kalendáře demonstruje efektivní propojení externích JavaScriptových knihoven s frameworkem Symfony v rámci moderní webové architektury.

```
const res = await fetch("{ path('app_calendar_event_create') }"), {
  method: "POST",
  headers: {
    "Content-Type": "application/json"
  },
  body: JSON.stringify({
    title: title,
    start: info.startStr,
    end: info.endStr,
    allDay: true
  })
});
```

**Obr. 21: Ukázka odeslání nové události kalendáře na server pomocí funkce fetch**

*Zdroj: Vlastní zpracování*

### 2.3.8 Přihlášení uživatelů a řízení přístupu

Přístup do aplikace je řízen autentizačním mechanismem frameworku Symfony, který zajišťuje identifikaci uživatelů prostřednictvím e-mailové adresy a hesla. Po odeslání přihlašovacího formuláře zpracovává požadavek vlastní autentizátor aplikace ověřující shodu údajů s databází. Systém následně generuje objekt typu Passport, jenž předává ověřenou identitu bezpečnostní komponentě frameworku pro vytvoření relace uživatele.

Proces registrace nového uživatele a jeho následné uložení do systému dokumentuje ukázka zdrojového kódu na obrázku 22. Algoritmus nejprve inicializuje novou instanci entity Users, do které jsou nastaveny základní atributy a příslušná role. Heslo je před zápisem bezpečně transformováno pomocí rozhraní UserPasswordHasherInterface. Finální fáze zahrnuje přiřazení uživatele ke konkrétní třídě a persistenci dat do databáze prostřednictvím nástroje ORM Doctrine s využitím metod persist() a flush.

```
$student = new User();

$student->setEmail($email);
$student->setRoles(['ROLE_STUDENT']);

$student->setPassword(
    $passwordHasher->hashPassword($student, $plainPassword)
);

$student->setRequirePasswordChange(true);

$student->setSchoolClass($class);

$em->persist($student);
$em->flush();
```

**Obr. 22: Vytvoření nového uživatele v roli žáka**

*Zdroj: Vlastní zpracování*

Aplikace striktně rozlišuje několik uživatelských rolí, konkrétně žáka, učitele, koordinátora a administrátora. Každá role disponuje přístupem výhradně k funkcím odpovídajícím přiděleným oprávněním, což zajišťuje bezpečné oddělení dat a funkcionalit systému. Bezpečnost citlivých údajů je podpořena ukládáním hesel v hashované podobě, k čemuž jsou využity vnitřní kryptografické mechanismy Symphony.

Při vytvoření nového žákovského účtu koordinátorem systém nastavuje uživateli dočasné heslo a současně aktivuje příznak vyžadující jeho změnu při prvním přihlášení. Po úspěšné autentizaci je uživatel v takovém případě automaticky přesměrován na formulář pro změnu hesla. Zmíněný mechanismus zvyšuje bezpečnost aplikace, protože administrativně přidělené heslo slouží pouze jako počáteční přístupový údaj.

## 2.4 Testování aplikace

Testování aplikace bylo zaměřeno na ověření správné funkčnosti jednotlivých modulů a na kontrolu použitelnosti UI. Cílem bylo potvrdit, že systém správně reaguje na běžné uživatelské operace a že jednotlivé části spolupracují bez chyb.

Proces probíhal průběžně během vývoje. Nejprve byly ověřovány izolované funkce, následně byla testována celková funkčnost při interakci různých typů uživatelů. Důraz byl kladen zejména na správnou práci s databází, kontrolu přístupových práv jednotlivých rolí a korektní zobrazování dat v rozhraní. Postupně byly prověřeny hlavní moduly: správa lekcí, tvorba kvízů, vyhodnocování výsledků, evidence docházky a práce s kalendářem.

### Ověřovány byly především následující funkce:

- autentizace uživatelů podle přidělené role,
- zobrazování výukových lekcí na základě ročníku,
- vytváření a uložení kvízů do databáze,
- vyhodnocování odpovědí a zápis výsledků jednotlivých pokusů,
- evidence docházky žáků,
- vizualizace výsledků a jejich následný export do formátu CSV,
- dynamická správa událostí v kalendáři.

Celé ověřování probíhalo formou lokálního spuštění ve vývojovém prostředí pomocí simulace běžné práce uživatele. Výsledky potvrdily plnou funkčnost všech implementovaných modulů.

### 2.4.1 Uživatelské testování

Součástí ověřování aplikace bylo uživatelské testování zaměřené na přehlednost a jednoduchost ovládání systému. Proces probíhal z pohledu tří klíčových rolí – žáka, učitele a koordinátora.

Žák se soustředil na interakci s výukovými lekciemi a odevzdávání kvízů. Bylo ověřeno, že uživatel dokáže snadno navigovat mezi lekciemi, zobrazit zadání a vyplnit připravený test. Po odeslání dat se v UI okamžitě zobrazí dosažený výsledek a celková úspěšnost, čímž je poskytnuta důležitá zpětná vazba.

Učitel testoval především správu docházky a monitoring výsledků. Ověření potvrdilo, že pedagog může plynule sledovat pokusy jednotlivých žáků a využívat funkci exportu do CSV souboru pro další zpracování dat. Koordinátor pak prověřil nástroje pro tvorbu obsahu, konkrétně přidávání nových lekcí a přípravu kvízových otázek pro jednotlivé hodiny.

Výsledky testování prokázaly, že aplikace disponuje intuitivním rozhraním, které zvyšuje celkové UX systému. Pro přehledné shrnutí byla vytvořena tabulka testovacích scénářů simulujících běžné situace.

**Tabulka č. 1 – Tabulka testů**

Testovaná funkce	Očekávaný výsledek	Skutečný výsledek
Přihlášení uživatele	Uživatel je přesměrován na odpovídající dashboard podle role	OK
Zobrazení lekcí	Student vidí lekce svého ročníku Učitel vidí všechny ročníky	OK OK
Vytvoření lekce	Lekce je vytvořena a uložena do databáze	OK
Upravení lekce	Změny jsou uloženy do databáze	OK
Vytvoření poznámky	Učitel může přidat poznámku	OK
Nastavení limitu	Učitel může nastavit časový limit pro vyplnění kvízu	OK
Otevření kvízu	Kvíz je zpřístupněn žákům k vyplnění	OK
Uzavření kvízu	Kvíz již není dostupný pro další pokusy	OK
Vytvoření kvízu	Kvíz a jeho otázky jsou uloženy do databáze	OK
Zobrazení výsledků	Uživatel vidí přehled výsledků kvízů	OK
Export výsledků	Výsledky jsou exportovány do CSV souboru	OK
Evidence docházky	Docházka žáků je uložena do databáze	OK
Přidání žáka	Nový žák je vytvořen a uložen v systému	OK
Přiřazení ke třídě	Žák je přiřazen ke zvolené třídě	OK
Přiřazení učitele	Učitel je přiřazen ke konkrétní třídě	OK
Kalendář	Událost je vytvořena a zobrazena v kalendáři	OK

*Zdroj: Vlastní zpracování*

Na základě provedených testů lze konstatovat, že jednotlivé funkce pracují správně a odpovídají očekávanému chování systému. Testování potvrdilo bezchybnou spolupráci všech modulů aplikace.

## 2.5 Nasazení aplikace

Vývoj a testování aplikace probíhalo v lokálním vývojovém prostředí postaveném na frameworku *Symfony 7.4.3* a programovacím jazyce *PHP 8.2.30*. Jako relační databázový systém byla použita databáze *MySQL*. Správa zdrojového kódu a editace projektu probíhala ve vývojovém prostředí *Visual Studio Code*.

Připojení aplikace k databázi je konfigurováno prostřednictvím systémového souboru `.env`, ve kterém je definována proměnná `DATABASE_URL`. Proměnná obsahuje nezbytné přístupové údaje (hostitel, port, uživatelské jméno a heslo) k databázovému serveru a název databáze.

Provoz aplikace zajišťuje lokální vývojový server *Symfony*. Po inicializaci serveru je webové rozhraní dostupné v prohlížeči na adrese:

`http://127.0.0.1:8001` (případně na portu určeném při startu).

### 2.5.1 Technický postup inicializace systému

Vzhledem k tomu, že je aplikace vyvíjena jako moderní webová platforma, vyžaduje její spuštění v lokálním prostředí správnou instalaci závislostí a přípravu databázového schématu. Následující kroky popisují základní operace potřebné k inicializaci systému:

1. Instalace PHP knihoven: Stažení všech potřebných komponent frameworku *Symfony* definovaných v souboru `composer.json` pomocí nástroje *Composer*:
  - `composer install`
2. Správa frontendových závislostí: Instalace balíčků *Node.js* a kompilace assetů (*CSS*, *JS*) pomocí nástroje *npm*:
  - `npm install`
  - `npm run dev` (pro vývojové prostředí) nebo `npm run build` (pro produkční sestavení).
3. Spuštění migrace databáze: Vytvoření základní struktury databáze *MySQL* pomocí nástroje *Doctrine*:
  - `php bin/console doctrine:migrations:migrate`
4. Spuštění lokálního serveru: Aktivace vývojového serveru pomocí *Symfony CLI*:
  - `symfony server:start`

Po úspěšném provedení doporučených kroků je systém připraven k provozu a dostupný na adrese specifikované v konzoli.

Pro demonstraci a testování byly v systému vytvořeny vzorové uživatelské účty reprezentující různé role v rámci školní hierarchie (administrátor, koordinátor, učitel, žák), viz tabulka 3. Testovací účty studentů mají tvar [student@test.cz](mailto:student@test.cz), [student1@test.cz](mailto:student1@test.cz) až [student5@test.cz](mailto:student5@test.cz) a pro všechny účty je nastaveno shodné heslo.

**Tab. 3: Přístupové údaje**

Role	Přihlašovací jméno	Heslo	Popis
Žák	<a href="mailto:student@test.cz">student@test.cz</a>	student123	Přístup k lekcím a kvízům
Učitel	<a href="mailto:teacher@test.cz">teacher@test.cz</a>	teacher123	Správa docházky a sledování výsledků
Koordinátor	<a href="mailto:coordinator@test.cz">coordinator@test.cz</a>	coordinator123	Správa lekcí a tvorba kvízů
Administrátor	<a href="mailto:admin@test.cz">admin@test.cz</a>	admin123	Správa uživatelů a systému

*Zdroj: Vlastní zpracování*

## 3 Možnost dalšího rozšíření aplikace

Navržená aplikace představuje funkční prototyp výukového systému, který je možné v budoucnu dále rozšiřovat o nové funkce a moduly. Díky zvolené modulární architektuře lze jednotlivé části systému postupně doplňovat a přizpůsobovat potřebám konkrétní školy nebo vyučujícího. Uvedené návrhy ilustrují potenciál systému pro další pedagogický i technologický rozvoj a ukazují možnosti jeho širšího využití ve školní praxi.

### 3.1 Rozšíření správy obsahu a poznámek

Další rozvoj systému v oblasti správy obsahu směřuje k rozšíření práce s multimediálními materiály a k lepší podpoře pedagogických poznámek učitele.

#### **Nativní multimediální obsah:**

Implementace nástrojů pro přímé vkládání a hostování souborů (videa, obrázky, pracovní listy) přímo v rámci infrastruktury aplikace. Cílem je úplná eliminace externích odkazů (např. na platformu YouTube) ve výukovém obsahu pro žáky. Navržené řešení má omezit nežádoucí „proklikávání“ na nerelevantní obsah a zvýšit bezpečnost i soustředění žáků na zadaný úkol.

#### **Diferenciace obsahu pro učitele:**

Externí metodické materiály a inspirační videa by byly nově dostupné výhradně v modulu učitelských poznámek. Zůstane tak zachována informační bohatost pro pedagoga, zatímco rozhraní pro žáka zůstane čisté, bezpečné a zaměřené na konkrétní cíle lekce.

#### **Integrovaná odevzdávací úkolů:**

Vytvoření vlastního modulu pro odevzdávání a správu žákovských prací přímo v rámci aplikace. Zanikla by nutnost využívat externí platformy (např. MS Teams) a výuka by se sjednotila „pod jednu střechu“, což usnadňuje kontrolu i následné hodnocení.

### 3.2 Interaktivita a optimalizace datového modelu

Možný vývoj systému zahrnuje rozšíření interaktivních výukových prvků a úpravy datového modelu podporující efektivnější správu kvízových otázek.

#### **Vlastní editor interaktivních cvičení:**

Implementace modulu pro tvorbu nativních vzdělávacích úloh (např. „doplňovačky“, „spojovačky“, „pexeso“ ...) přímo v administraci aplikace. Žáci by tato cvičení plnili bezprostředně před kvízem za účelem upevnění znalostí.

#### **Převedení správy kvízů na samostatné databázové entity:**

Současné ukládání otázek ve formátu JSON, které zajišťuje flexibilitu prototypu, by bylo v budoucnu nahrazeno samostatnými databázovými entitami. Zmíněné řešení by vytvořilo „banku otázek“ pro opětovné využití v různých kvízech a umožnilo detailnější statistické vyhodnocování úspěšnosti žáků.

### 3.3 Monitoring a organizační funkce

Další rozvoj může směřovat také k nástrojům pro sledování studijní aktivity žáků a k podpoře organizace výuky prostřednictvím analytických funkcí.

#### **Pokročilý monitoring pokroku:**

Vývoj analytického modulu pro sledování celkové aktivity žáka v lekcích i kvízech, který učitelé poskytne detailní přehled o individuálním postupu každého žáka ve výuce.

#### **Inteligentní kalendář a notifikace:**

Rozšíření funkcí kalendáře o systém připomínek a plánování aktivit s přímou vazbou na konkrétní lekce. Notifikační systém by uživatele automaticky informoval o nových úkolech či blížících se termínech.

### 3.4 Inkluze a škálovatelnost

Rozvoj systému může zahrnovat úpravy podporující přístupnost aplikace pro žáky se speciálními vzdělávacími potřebami a současně umožnit rozšíření obsahu pro vyšší ročníky.

#### **Inkluze a bezbariérovost:**

Úprava uživatelského rozhraní aplikace s ohledem na potřeby žáků se speciálními vzdělávacími potřebami. Jedná se například o úpravu kontrastu, podporu pro žáky s dyslexií nebo přehlednější strukturu rozhraní. Cílem je zvýšit přístupnost systému a zajistit rovnocenný přístup ke vzdělávacím materiálům.

#### **Vertikální rozšíření obsahu:**

Adaptace a rozšíření výukového obsahu pro vyšší ročníky základní školy, což by vyžadovalo implementaci komplexnějších programátorských úloh a pokročilejších logických struktur.

## Závěr

Cílem bakalářské práce byl návrh a implementace webové aplikace podporující výuku informatiky na prvním stupni základní školy, která usnadňuje přípravu vyučovacích hodin učitelům i zastupujícím pedagogům. Realizace vycházela z aktuálních požadavků RVP ZV a zohledňovala vývojové i kognitivní zvláštnosti žáků mladšího školního věku.

V rámci praktické části byla vyvinuta aplikace umožňující správu výukových lekcí, tvorbu a vyplňování kvízů, sledování výsledků, evidenci docházky a plánování aktivit prostřednictvím kalendáře. Systém je postaven na frameworku Symfony a využívá relační databázi pro ukládání dat o uživatelích, lekcích a výsledcích pokusů.

Přínos práce spočívá především v propojení pedagogických principů s realizací konkrétního softwarového řešení. Vytvořená platforma nabízí jednotnou strukturu hodin, přehledné UI a metodickou podporu pro učitele bez hlubší specializace v oblasti informatiky. Současně poskytuje nástroje pro sledování pokroku žáků a organizaci výuky v digitálním prostředí, čímž přispívá ke zlepšení uživatelské zkušenosti všech zúčastněných stran.

Výsledky práce jsou využitelné v běžné školní praxi jako podpůrný nástroj při výuce informatiky. Další rozvoj aplikace může směřovat k rozšíření obsahu pro vyšší ročníky, doplnění pokročilé analytiky pokroku žáků nebo k adaptaci systému pro potřeby speciálního vzdělávání. Díky modulární architektuře je možné systém dále rozšiřovat a přizpůsobovat specifickým potřebám jednotlivých škol. Navržené řešení tak představuje stabilní základ pro budoucí vývoj digitálních nástrojů podporujících moderní výuku.

Při přípravě některých částí textu a při konzultaci programátorských postupů byl využit jazykový model ChatGPT (OpenAI) jako podpůrný nástroj pro formulaci textu a vysvětlení technických principů.

## Seznam použité literatury

- BOURGEOIS, Ania; BIRCH, Peter a DAVYDOVSKAIA, Olga. *Digital Education at School in Europe* [online]. European Commission, 2019 [cit. 2025-11-23].  
Dostupné z: [https://www.dzs.cz/sites/default/files/2023-01/Eurydice\\_Brief\\_Digital\\_Education\\_at\\_School\\_in\\_Europe.pdf](https://www.dzs.cz/sites/default/files/2023-01/Eurydice_Brief_Digital_Education_at_School_in_Europe.pdf)
- HLÁSNÁ, Pavla; KLÍMOVÁ, Blanka a PAULOVÁ, Petra. *Use of Information and Communication Technologies in Primary Education — A Case Study of the Czech Republic* [online]. Academia.edu, 2017 [cit. 2025-11-23].  
Dostupné z: [https://www.academia.edu/124706560/Use\\_of\\_Information\\_and\\_Communication\\_Technologies\\_in\\_Primary\\_Education\\_A\\_Case\\_Study\\_of\\_the\\_Czech\\_Republic](https://www.academia.edu/124706560/Use_of_Information_and_Communication_Technologies_in_Primary_Education_A_Case_Study_of_the_Czech_Republic)
- HREJSEMNOU, Pavel. *Psychologie barev marketingu*. IDEAIFY, 2025. ISBN 978-80-53028-26-4.
- LEMROVÁ, Soňa. *Pedagogická psychologie*. Vydání 1. Praha: Grada, 2024. 220 stran; Pedagogika. ISBN 978-80-271-1713-0.
- MÁČA, Jindřich. *Úvod do Symfony frameworku pro PHP* [online]. Itnetwork.cz [cit. 2025-11-22].  
Dostupné z: <https://www.itnetwork.cz/php/symfony/zaklady/uvod-do-symfony-frameworku-pro-php>
- NORMAN, Donald A. *Design pro každý den*. 1. vyd. v českém jazyce. Praha: Dokořán, 2010. 271 s. ISBN 978-80-7363-314-1.
- NPI. *Přípravenost škol na rozvoj digitálních kompetencí* [online]. Národní pedagogický institut České republiky, 2024 [cit. 2025-11-23].  
Dostupné z: <https://digitalizace.rvp.cz/files/ucitele-zsag-final-111.pdf>
- NPI. *Rámcový vzdělávací program pro základní vzdělávání, část C – Informatika* [online]. Národní pedagogický institut ČR, 2021 [cit. 2025-11-23].  
Dostupné z: <https://digitalizace.rvp.cz/files/informatika-2021.pdf>
- OECD. *OECD Digital Education Outlook 2023* [online]. OECD, 2023 [cit. 2025-11-23].  
Dostupné z: [https://www.oecd.org/en/publications/oecd-digital-education-outlook-2023\\_c74f03de-en.html](https://www.oecd.org/en/publications/oecd-digital-education-outlook-2023_c74f03de-en.html)
- ORACLE CORPORATION. *Manuál MySQL Workbench* [online]. Oracle, 2025 [cit. 2025-11-23].  
Dostupné z: [MySQL :: Manuál MySQL Workbench :: 9.1.1.3 EER diagramy](#)
- STARK, Oskar. *Databases and the Doctrine ORM* [online]. Symfony, 2025 [cit. 2025-11-23].  
Dostupné z: <https://symfony.com/doc/current/doctrine.html>
- THE PHP GROUP. *PHP Manual* [online]. The PHP Group, 2025 [cit. 2025-11-23].  
Dostupné z: <https://www.php.net/manual/en/>
- VÁGNEROVÁ, Marie. *Vývojová psychologie: dětství, dospělost, stáří*. Vyd. 1. Praha: Portál, 2000. 522 s. ISBN 80-7188-308-0.
- W3C. *CSS Snapshot 2023* [online]. World Wide Web Consortium, 2023 [cit. 2025-11-23].  
Dostupné z: <https://www.w3.org/TR/css-2023/>

W3SCHOOLS. *HTML Tutorial* [online]. W3Schools, 2025 [cit. 2025-11-23]. Dostupné z: [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp).

W3SCHOOLS. *JavaScript Tutorial* [online]. W3Schools, 2025 [cit. 2025-11-23]. Dostupné z: <https://www.w3schools.com/js/DEFAULT.asp>