

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**  
**FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-16605-111184

**MATEMATICKÝ TRENAŽÉR**  
**BAKALÁRSKA PRÁCA**

**2025**

**Bence Bodnár**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**  
**FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-16605-111184

**MATEMATICKÝ TRENAŽÉR**  
**BAKALÁRSKA PRÁCA**

Študijný program: Aplikovaná informatika  
Názov študijného odboru: Informatika  
Školiace pracovisko: Ústav informatiky a matematiky  
Vedúci záverečnej práce: doc. RNDr. Oľga Nánásiová, PhD.

**Bratislava 2025**

**Bence Bodnár**

# SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Aplikovaná informatika
Autor:	Bence Bodnár
Bakalárska práca:	Matematický trenažér
Vedúci záverečnej práce:	doc. RNDr. Olga Nánásiová, PhD.
Miesto a rok predloženia práce:	Bratislava 2025

V tejto bakalárskej práci sa zaoberáme vývojom trojvrstvovej webovej aplikácie zameranej na e-learning matematiky, konkrétne pravdepodobnosti a štatistiky. Cieľom práce bolo navrhnúť a implementovať užívateľsky orientovaný frontend pomocou Angular frameworku, pričom sú využívané knižnice Bootstrap a Material UI na zabezpečenie intuitívneho rozhrania. Na druhej strane, backend aplikácie bol vyvinutý pomocou Node.js a frameworku Next.js s cieľom poskytnúť efektívne spracovanie dát a logiky aplikácie. S PostgreSQL databázou sme pracovali na ukladaní a spracovaní užívateľských dát a obsahu. Celá aplikácia je nakoniec nasadená v Docker kontajneroch, čo umožňuje jednoduchšiu distribúciu a nasadenie aplikácie. Výsledkom je komplexná e-learningová platforma, ktorá umožňuje študentom testovať svoje znalosti prostredníctvom testov, úloh a študijných materiálov, a tiež analyzovať ich pokrok a vývoj. Tento projekt predstavuje dôležitý krok smerom k moderným pedagogickým metódam, ktoré využívajú technologické inovácie na zlepšenie vzdelávania.

Kľúčové slová: Docker, PostgreSQL, Framework, Node.js, Angular, Pravdepodobnosť

# ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Applied Informatics
Author:	Bence Bodnár
Bachelor's thesis:	Mathematics trainer
Supervisor:	doc. RNDr. Oľga Nánásiová, PhD.
Place and year of submission:	Bratislava 2025

In this bachelor thesis we are developing a three-layer web application focused for e-learning mathematics, specifically probability and statistics. The aim of the work was to design and implement a user-oriented frontend using the Angular framework, using the Bootstrap and Material UI libraries to provide an intuitive editing. On the other hand, the backend of the application was developed using Node.js and the framework Next.js framework in order to provide efficient data processing and application logic. With PostgreSQL database, we worked on storing and processing user data and content. The entire appli- Finally, the entire application is deployed in Docker containers, which allows for easier distribution and deployment of the application. The result is a comprehensive e-learning platform that enables learners to test their knowledge through tests, assignments and study materials, and also analyse their progress and development. This project represents an important step towards modern pedagogical methods that use technological innovation to improve education.

Keywords: Docker, PostgreSQL, Framework, Node.js, Angular, Probability

# Podakovanie

Podakovanie patrí mojej školiteľke doc. RNDr. Oľga Nánásiová, PhD. za poskytnutie poznatkov z oblasti, odborné konzultácie a čas, ktorý mi venovala pri vypracovaní mojej záverečnej práce.

# Zoznam skratiek

<b>AJAX</b>	Asynchronous JavaScript and XML
<b>API</b>	Rozhranie pre programovanie aplikácií
<b>CSS</b>	Kaskádové štýly (Cascading Style Sheets)
<b>DOM</b>	Document Object Model
<b>HTML</b>	Hypertextový značkovací jazyk (HyperText Markup Language)
<b>SCSS</b>	Sassy CSS
<b>UI</b>	Užívateľské rozhranie (User Interface)
<b>UX</b>	Užívateľský zážitok (User Experience)

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Analýza</b>	<b>2</b>
1.1 Brilliant.org . . . . .	2
1.2 Khan Academy . . . . .	2
1.3 Vieme matiku . . . . .	3
1.4 Zhodnotenie . . . . .	3
<b>2 Použité technológie a knižnice</b>	<b>4</b>
2.1 Frontend . . . . .	4
2.1.1 HTML . . . . .	4
2.1.2 CSS . . . . .	5
2.1.3 SCSS . . . . .	5
2.1.4 JavaScript . . . . .	5
2.1.5 TypeScript . . . . .	6
2.1.6 MathJax . . . . .	6
2.2 Backend . . . . .	6
2.2.1 CORS . . . . .	7
2.2.2 Node.js . . . . .	7
2.2.3 Express.js . . . . .	7
2.3 Databázové systémy . . . . .	8
2.3.1 PostgreSQL . . . . .	8
2.3.2 DBDiagram . . . . .	8
2.4 Framework . . . . .	8
2.4.1 Frontendové frameworky . . . . .	8
2.4.2 Backendové frameworky . . . . .	8
2.5 UI a UX . . . . .	8
2.5.1 Figma . . . . .	8
2.5.2 Material UI . . . . .	9
2.5.3 Angular . . . . .	9
2.6 Gamifikácia . . . . .	9
2.6.1 Gamifikácia v e-learningu . . . . .	9
2.7 Server . . . . .	10
2.7.1 Kontejnerizácia . . . . .	10

2.7.2	Docker . . . . .	10
2.7.3	NGINX . . . . .	10
2.8	GIT . . . . .	10
<b>Záver</b>		<b>11</b>
<b>Zoznam použitej literatúry</b>		<b>12</b>



# Zoznam obrázkov a tabuliek

Tabuľka 1	Vzdelávacie platformy . . . . .	4
-----------	---------------------------------	---

# Zoznam algoritmov

# Zoznam výpisov

# Úvod

Štúdium matematickej štatistiky a pravdepodobnosti je kritické pre porozumenie a analyzovanie náhodných javov a dát v rôznych oblastiach, ako napríklad v ekonómii, vedeckom výskume alebo medicíne. Tieto oblasti matematiky zahŕňajú širokú škálu tém, vrátane modusu, mediánu, stredovej hodnoty, náhodných premenných, kombinatoriky a podmienenej pravdepodobnosti. Každá z týchto tém poskytuje unikátne nástroje na kvantifikáciu a analýzu dátových súborov, čo je kľúčové pre predpovedanie a porozumenie rôznym javom a trendom. S cieľom podporiť systematické vzdelávanie v týchto dôležitých oblastiach sme sa rozhodli vyvinúť trojvrstvovú webovú aplikáciu. Táto aplikácia, využívajúca moderné technológie ako Angular framework s knižnicami Bootstrap a Material UI pre frontend, a Node.js s frameworkom Next.js pre backend, sa zameriava na poskytovanie interaktívnych učebných materiálov, testov a študijných materiálov v týchto matematických témach. Cieľom tejto práce je nielen uľahčiť proces učenia sa matematickej štatistiky a pravdepodobnosti, ale aj zlepšiť zrozumiteľnosť a prístupnosť týchto konceptov pre študentov. Naša aplikácia má ambíciu prispieť k zvýšeniu efektivity vzdelávacieho procesu v tejto dôležitej matematickej oblasti a poskytnúť študentom moderný a efektívny nástroj na zlepšenie ich matematických schopností a analytického myslenia.

# 1 Analýza

V tejto kapitole sa venujeme rozboru dostupných platforiem pre e-learning matematiky. Cieľom je identifikovať platformy, porovnať ich funkcie a odhaliť medzery, ktoré naša webová aplikácia môže vyplniť. Na trhu existuje široká škála platforiem pre e-learning rôznych matematických tém, z ktorých každá ponúka rôzne riešenia, funkcie a zameriava sa na odlišné cieľové skupiny.

## 1.1 Brilliant.org

Brilliant.org je online vzdelávacia platforma zameraná na interaktívne kurzy v oblastiach matematiky, vedy a počítačovej vedy. Je navrhnutá tak, aby podporovala aktívne učenie prostredníctvom riešenia problémov a interaktívnych výziev, čím pomáha študentom rozvíjať kritické myslenie a logické schopnosti. Platforma ponúka viac ako 60 kurzov, ktoré sú prispôsobené rôznym úrovňam znalostí, od začiatočníkov po pokročilých.

Medzi jej hlavné výhody patria interaktívne lekcie, ktoré sú navrhnuté tak, aby boli pútavé a vyžadovali aktívnu účasť študentov, čím zvyšujú efektivitu učenia. Umožňuje tiež flexibilné a samostatné štúdium, čo je ideálne pre individuálne potreby. Platforma ponúka denné výzvy na rôzne témy, ktoré pomáhajú udržiavať študentov motivovaných a neustále zapojených do procesu učenia. Nevýhodou je, že táto platforma je platená a dostupná len v anglickom jazyku, čo môže predstavovať prekážku pre niektorých študentov. [1]

## 1.2 Khan Academy

Táto platforma ponúka bezplatné videokurzy a interaktívne cvičenia z rôznych oblastí matematiky, vrátane vysokoškolskej štatistiky a pravdepodobnosti. Je vhodná pre študentov základných aj vysokých škôl. Medzi jej výhody patrí široká škála obsahu, jednoduché použitie a dostupnosť pre rôzne úrovne znalostí. Dostupné zdroje k daným témam sú prehľadné a dobre štruktúrované. Taktiež ponúka možnosť sledovania pokroku, získavania bodov a odznakov za splnené kapitoly, čím motivuje študentov k učeniu prostredníctvom gamifikácie<sup>1</sup>. Nevýhodou je, že je dostupná len v anglickom jazyku, čo môže byť pre niektorých študentov prekážkou. Používateľské rozhranie môže byť z dôvodu množstva obsahu pre niektorých používateľov neprehľadné, najmä ak sa na platforme nachádzajú prvýkrát. Napriek týmto nedostatkom je platforma považovaná za jeden z najlepších nástrojov na online vzdelávanie a sebarozvoj. [2]

---

<sup>1</sup>Gamifikácia je využitie herných prvkov a mechaník v nehermom prostredí s cieľom zvýšiť motiváciu, zapojenie a efektivitu používateľov.

## 1.3 Vieme matiku

Najpopulárnejším slovenským portálom pre e-learning matematiky je Vieme matiku. Táto platforma ponúka rôzne kurzy a cvičenia z matematiky pre žiakov základných a stredných škôl. Medzi jej výhody patrí dostupnosť pre slovenských žiakov, široký výber tém, rôzne formy precvičovania, do ktorých patrí grafické znázornenie úloh a možnosť sledovania pokroku. Ponúka taktiež hravé prvky, ako sú grafické a zvukové efekty, ktoré môžu zvýšiť motiváciu žiakov. Vyznačuje sa taktiež jednoduchým použitím a prehľadným rozhraním. Nevýhodou je, že nie je dostupná pre študentov mimo Slovenska, je podporovaná len v slovenčine. Platforma slúži na precvičovanie matematických úloh, ale neponúka zdroje pre samostatné štúdium alebo nápovedy. Zároveň, v prípade, že by sme chceli naplno využiť všetky jej funkcie, by bolo potrebné si zakúpiť licenciu. [3]

## 1.4 Zhodnotenie

Počas analýzy existujúcich vzdelávacích platforiem sme zistili, že na trhu chýbajú lokalizované a cenovo dostupné e-learningové riešenia pre stredoškolských a vysokoškolských študentov, ktoré by efektívne kombinovali gamifikáciu, interaktivitu a prehľadné rozhranie. Existujúce platformy, ako Brilliant.org a Khan Academy, ponúkajú kvalitné vzdelávacie materiály, ale ich dostupnosť je limitovaná anglickým jazykom a v prípade Brilliant.org aj plateným modelom. Vieme Matiku síce poskytuje lokalizovaný obsah, ale nezohľadňuje pokročilé potreby samostatného štúdia a je obmedzená na úzky okruh používateľov. Analyzované platformy ukázali širokú škálu prístupov, pričom mnohé sa zameriavajú na riešenie komplexných úloh alebo tradičné formy vzdelávania. Tieto prístupy však často nekladú dôraz na intuitívne osvojovanie matematických konceptov a podporu samostatného učenia. Tieto poznatky nám umožňujú identifikovať medzery a formulovať jasné požiadavky na vývoj novej aplikácie, ktorá by ponúkala lokalizovaný obsah, interaktívne učenie a dostupnosť pre rôzne cieľové skupiny.

Tabuľka 1: Vzdelávacie platformy

Platforma	Funkcie	Cieľová skupina	Cena
Khan Academy	Videokurzy, cvičenia	Všetky úrovne	Bezplatná
Brilliant.org	Gamifikované kurzy	Stredné a Vysoké školy	Platená
Vieme Matiku	Online kurzy matematiky	Základné a Stredné školy	Čiastočne bezplatná

## 2 Použité technológie a knižnice

V tejto kapitole sa podrobne venujeme technológiám a knižniciam, ktoré plánujeme použiť na vývoj webovej aplikácie pre e-learning matematickej štatistiky a pravdepodobnosti. Výber technológií je založený na princípoch flexibility, kompatibility, bezpečnosti a aktívnej komunity vývojárov.

### 2.1 Frontend

Frontend je časť softvérového vývoja, ktorá sa zaoberá tým, čo používateľ vidí a s čím interaguje pri práci s aplikáciou alebo webovou stránkou. Ide o viditeľnú vrstvu aplikácie, ktorá zahŕňa všetky prvky používateľského rozhrania (UI) a je priamo zodpovedná za používateľskú skúsenosť (UX).

V kontexte nášho webového vývoja predstavuje frontend technológie a nástroje používané na tvorbu webových stránok, ktoré sú dostupné a vykresľované v internetových prehliadačoch. Zahŕňa návrh, implementáciu a optimalizáciu používateľského rozhrania tak, aby bolo esteticky príťažlivé, funkčné a dostupné na rôznych zariadeniach a platformách.

#### 2.1.1 HTML

Hypertextový značkovací jazyk (HyperText Markup Language) (HTML) je značkovací jazyk používaný na tvorbu a štruktúrovanie obsahu webových stránok. Umožňuje definovať rôzne prvky, ako sú nadpisy, odseky, obrázky či odkazy, čím určuje základnú kostru a vzhľad webovej stránky. Napriek častým mylným predstavám, HTML nie je programovací jazyk, keďže neumožňuje vytvárať podmienené logické operácie alebo funkcie. Jeho hlavnou úlohou je prezentácia a organizácia obsahu pre webové prehliadače. [4]

### 2.1.2 CSS

Kaskádové štýly (Cascading Style Sheets) (CSS) [5] je štýlovací jazyk používaný na definovanie vzhľadu a formátovania webových stránok. Umožňuje oddeliť vizuálnu prezentáciu od štruktúry obsahu definovanej v HTML, čím zjednodušuje údržbu a aktualizáciu dizajnu. Pomocou CSS je možné nastaviť rôzne vizuálne vlastnosti, ako sú farby, písma, veľkosti, rozloženie prvkov a ďalšie aspekty dizajnu. Taktiež podporuje tvorbu responzívnych dizajnov, ktoré sa prispôbujú rôznym zariadeniam a veľkostiam obrazoviek. Moderné techniky, ako flexbox a grid, umožňujú presné rozmiestnenie a zarovnanie prvkov na stránke, čo je užitočné pri tvorbe komplexných rozložení.

### 2.1.3 SCSS

Sassy CSS (SCSS) [6] je rozšírenie jazyka CSS, ktoré pridáva pokročilé funkcie pre efektívnejšie štýlovanie webových stránok. SCSS umožňuje používať premenné, vnáranie selektorov, mixiny, funkcie a operácie, čím zjednodušuje správu a údržbu štýlov. Vďaka svojim vlastnostiam podporuje modulárny prístup k tvorbe štýlov, čím zlepšuje čitateľnosť kódu a urýchľuje vývoj.

SCSS používa štandardnú CSS syntax s doplnením nových funkcií, čo zabezpečuje spätnú kompatibilitu. Kód napísaný v SCSS sa následne kompiluje do klasického CSS, ktoré podporujú všetky moderné prehliadače. Tento proces zvyšuje flexibilitu vývoja a umožňuje tvorbu komplexných štýlových štruktúr.

### 2.1.4 JavaScript

JavaScript [7] je interpretovaný programovací jazyk ktorý umožňuje dynamickú interakciu s používateľom a zmeny obsahu webových stránok bez nutnosti ich opätovného načítania.

Podporuje objektovo orientované programovanie s triedami, objektmi a metódami, čo umožňuje tvorbu komplexných aplikácií. Vďaka svojej dynamickej povahe dokáže meniť obsah a štruktúru stránky počas jej behu.

Medzi jeho funkcie patrí funkcionálne programovanie, kde sú funkcie považované za prvotriedne objekty, a programovanie riadené udalosťami, ktoré umožňuje reagovať na interakcie používateľa, napríklad na kliknutia.

Je multiplatformový a podporuje rôzne zariadenia, ako sú počítače, smartfóny a tablety. Populárne knižnice a rámce ako jQuery, React, Angular a Vue výrazne uľahčujú vývoj aplikácií.

Medzi hlavné vlastnosti patrí manipulácia s Document Object Model (DOM), spraco-



vanie udalostí, manipulácia s dátami a podpora asynchrónnych volaní na server pomocou techniky Asynchronous JavaScript and XML (AJAX). Tieto vlastnosti z neho robia základný nástroj na tvorbu moderných webových aplikácií.

### 2.1.5 TypeScript

TypeScript [8] je programovací jazyk vyvinutý spoločnosťou Microsoft, ktorý rozširuje možnosti JavaScriptu pridaním statického typovania a pokročilých objektovo orientovaných prvkov. Tým umožňuje vývojárom identifikovať chyby už počas vývoja, čo zvyšuje spoľahlivosť a udržiavateľnosť kódu. TypeScript je nadmnožinou JavaScriptu, čo znamená, že všetok platný kód v JavaScripte je kompatibilný s TypeScriptom. Po napísaní sa kód v JavaScripte transpiluje do štandardného JavaScriptu, ktorý je podporovaný vo všetkých moderných prehliadačoch. Tento prístup umožňuje využívať výhody moderných programovacích techník pri zachovaní širokej kompatibility a flexibility, ktorú JavaScript ponúka.

### 2.1.6 MathJax

MathJax [9] je open-source JavaScriptový engine určený na zobrazovanie matematickej notácie, ako sú LaTeX, MathML a AsciiMath, v moderných webových prehliadačoch. Je navrhnutý tak, aby konsolidoval pokroky vo webových technológiách do jednotnej platformy pre matematiku na webe, podporujúc hlavné prehliadače a operačné systémy, vrátane mobilných zariadení. Používatelia nemusia inštalovať žiadne doplnky ani softvér; stačí, aby autor stránky zahrnul MathJax a matematický obsah do webovej stránky, a MathJax sa postará o zvyšok.

## 2.2 Backend

Backend predstavuje časť softvérovej aplikácie, ktorá nie je priamo prístupná používateľom; ide o serverovú stranu v klient-server architektúre. Zodpovedá za hlavnú funkcionálnosť aplikácie, vrátane spracovania webových požiadaviek, manipulácie s dátami a ich ukladania v databázach. Backend spolupracuje s frontendom, ktorý tvorí prezentačnú vrstvu aplikácie, s cieľom zabezpečiť komplexnú používateľskú skúsenosť. Dáta generované na backende sú odosielané na frontend, kde sú prezentované používateľovi. Hoci sú často backend a frontend vyvíjané oddelenými tímami, hranica medzi nimi môže byť nejasná, čo vedie k prístupu známemu ako full-stack<sup>2</sup> development, kde programátori pracujú na oboch stranách

---

<sup>2</sup>Full-stack vývoj je proces, kde vývojár pracuje na frontende aj backende, teda na používateľskom rozhraní aj serverovej logike.

aplikácie. V minulosti backend pozostával prevažne z jednoduchých serverových skriptov, avšak s rozvojom webových technológií sa využívajú pokročilé frameworky umožňujúce dynamickú generáciu obsahu. Efektívny backendový kód je kľúčový pre optimalizáciu výkonu aplikácie, minimalizáciu zaťaženia servera a databázy, a zabezpečenie rýchlej odozvy pre používateľov. [10]

### 2.2.1 CORS

CORS (Cross-Origin Resource Sharing) predstavuje mechanizmus, ktorý umožňuje realizáciu HTTP požiadaviek medzi rôznymi doménami (tzv. cross-origin požiadavky). Ide o spôsob, ako prehliadače môžu bezpečne vykonávať požiadavky na zdroje umiestnené na iných serveroch, než odkiaľ pochádza pôvodná webová stránka. V minulosti takéto požiadavky bránila politika rovnakého pôvodu (same-origin policy), ktorá obmedzovala skripty bežiacie v prehliadači výhradne na komunikáciu so serverom tej istej domény.

Mechanizmus CORS funguje tak, že server explicitne definuje, kto môže vytvárať požiadavky na jeho API <sup>3</sup>, aké typy požiadaviek sú povolené a aké HTTP metódy sú podporované. Toto sa realizuje prostredníctvom špeciálnych HTTP hlavičiek, ktoré umožňujú prehliadaču určiť, či konkrétna požiadavka medzi rôznymi doménami môže byť bezpečne vykonaná. [11]

### 2.2.2 Node.js

Node.js je runtime prostredie umožňujúce použitie JavaScriptu na strane servera, pričom využíva flexibilitu a jednoduchosť tohto programovacieho jazyka. Výhodou JavaScriptu sú jeho pokročilé koncepty ako „first-class functions“ (funkcie prvej triedy) a closures, ktoré umožňujú vytváranie efektívnych webových aplikácií. Hoci JavaScript býva kritizovaný za nespoľahlivosť, táto kritika vyplýva predovšetkým zo zvláštností DOM-u v prehliadačoch, nie zo samotného jazyka. Node.js tak využíva práve dobre definované vlastnosti JavaScriptu, ktoré umožňujú vytvárať vysoko výkonné platformy pre webové aplikácie. [12] <sup>4</sup>

### 2.2.3 Express.js

Express je JavaScriptový framework fungujúci ako ľahká nadstavba nad Node.js, ktorý zjednodušuje tvorbu webových aplikácií a API. Zatiaľ čo Node.js poskytuje základné

---

<sup>3</sup>API (Application Programming Interface) je rozhranie, ktoré umožňuje komunikáciu a výmenu dát medzi rôznymi softvérovými aplikáciami alebo systémami.

<sup>4</sup>Runtime prostredie je prostredie poskytujúce aplikácii všetky potrebné zdroje, knižnice a služby, ktoré sú potrebné na jej spustenie a beh.

prostredie pre spúšťanie JavaScriptu na serveri, Express pridáva štruktúru a pomáha vývojárom vyhnúť sa opakovanému a zdĺhavému písaniu nízkoúrovňového kódu. Express je teda nástroj, ktorý zjednodušuje a zrýchľuje vývoj tým, že poskytuje vyššiu úroveň abstrakcie oproti základným funkciám dostupným v čistom Node.js. [13]

## **2.3 Databázové systémy**

### **2.3.1 PostgreSQL**

### **2.3.2 DBDiagram**

## **2.4 Framework**

### **2.4.1 Frontendové frameworky**

### **2.4.2 Backendové frameworky**

## **2.5 UI a UX**

Používateľské rozhranie predstavuje vizuálnu časť digitálneho produktu, s ktorou používateľ priamo interaguje. Zahŕňa prvky ako tlačidlá, ikony, typografiu a farebné schémy. Cieľom UI dizajnu je vytvoriť esteticky príjemné a funkčné prostredie, ktoré uľahčuje používateľovi navigáciu a interakciu s produktom.

Používateľský zážitok sa zameriava na celkovú skúsenosť používateľa pri interakcii s produktom alebo službou. Ide o to, ako intuitívne a efektívne dokáže používateľ dosiahnuť svoje ciele. UX dizajn zahŕňa analýzu potrieb používateľov, návrh informačnej architektúry, prototypovanie a testovanie, s cieľom zabezpečiť, aby bol produkt nielen funkčný, ale aj príjemný na používanie.

Hoci sú UI a UX odlišné disciplíny, úzko spolupracujú s cieľom vytvoriť digitálne produkty, ktoré sú nielen vizuálne atraktívne, ale aj používateľsky prívetivé a efektívne.[14]

### **2.5.1 Figma**

Figma je cloudová platforma na návrh a prototypovanie používateľských rozhraní pre webové a mobilné aplikácie. Umožňuje tímovú spoluprácu v reálnom čase prostredníctvom webového prehliadača alebo desktopovej aplikácie. Podporuje vektorové ilustrácie, interaktívne prototypy a dizajnové systémy. Všetky projekty sú uložené v cloude, čo zjednodušuje prístup a zdieľanie. [15]

### 2.5.2 Material UI

Angular Material UI je oficiálna knižnica komponentov pre Angular, ktorá implementuje dizajnové princípy Material Design od spoločnosti Google. Poskytuje predpripravené a prispôsobiteľné komponenty, ako sú tlačidlá, formuláre, tabuľky či navigačné panely, ktoré umožňujú vývojárom rýchlo vytvárať moderné, responzívne a esteticky príjemné používateľské rozhrania. Angular Material UI podporuje integráciu s Angular frameworkom a zabezpečuje konzistentný dizajn a vysokú úroveň použiteľnosti v rámci aplikácií. [16]

### 2.5.3 Angular

Angular je open-source framework vyvinutý spoločnosťou Google, určený na tvorbu dynamických a responzívnych webových aplikácií. Umožňuje vývojárom vytvárať aplikácie s bohatou funkcionalitou prostredníctvom komponentovo orientovanej architektúry, ktorá podporuje opätovné použitie kódu a zjednodušuje údržbu aplikácií. Angular využíva TypeScript, nadmnožinu JavaScriptu, ktorá pridáva statické typovanie a ďalšie funkcie zlepšujúce vývojový proces. Medzi kľúčové vlastnosti Angularu patrí obojsmerná väzba dát, ktorá synchronizuje model a zobrazenie, a modulárny systém, ktorý umožňuje rozdelenie aplikácie na menšie, ľahko spravovateľné časti. Angular tiež obsahuje nástroje na správu formulárov, komunikáciu so serverom a smerovanie, čo umožňuje vytvárať komplexné aplikácie s minimálnym úsilím. [17]

## 2.6 Gamifikácia

Gamifikácia je aplikácia herných prvkov a princípov v neherných kontextoch s cieľom zvýšiť angažovanosť a motiváciu jednotlivcov pri vykonávaní určitých aktivít. Tento prístup sa využíva v rôznych oblastiach, ako sú vzdelávanie, marketing, podnikanie a osobný rozvoj. Vo vzdelávacom prostredí to znamená implementáciu mechanizmov, ako sú zbieranie bodov, získavanie odmien, porovnávanie sa s ostatnými či postupovanie na vyššie úrovne, aby sa proces učenia stal interaktívnejším a pútavejším.

### 2.6.1 Gamifikácia v e-learningu

Gamifikácia v e-learningu zahŕňa integráciu herných prvkov, ako sú body, odznaky, rebríčky a úrovne, do vzdelávacích online prostredí s cieľom zvýšiť motiváciu a angažovanosť študentov. Podľa príspevku v zborníku z medzinárodnej vedeckej konferencie "Vzdělávání dospělých 2021"[18] je gamifikácia efektívnym nástrojom na podporu aktívneho učenia, pretože využíva prirodzenú ľudskú tendenciu k hre a súťaživosti. Implementácia herných

mechanizmov v e-learningových kurzoch môže viesť k zlepšeniu zapojenia študentov, zvýšeniu ich motivácie a následne k lepším vzdelávacím výsledkom.

## **2.7 Server**

### **2.7.1 Kontejnerizácia**

### **2.7.2 Docker**

### **2.7.3 NGINX**

## **2.8 GIT**

# Záver

Conclusion is going to be where?

Here.

# Zoznam použitej literatúry

1. *Brilliant* [online]. Brilliant. [cit. 2024-12-08]. Dostupné z : <https://www.brilliant.org/>.
2. *Khan Academy* [online]. Khan Academy. [cit. 2024-12-08]. Dostupné z : <https://www.khanacademy.org/>.
3. *Vieme matiku* [online]. Vieme to. [cit. 2024-12-08]. Dostupné z : <https://www.viemeto.org/zhrnutie-projektu>.
4. DVOŘÁK, Jan. *Čo to je jazyk HTML a jeho základy*. 2022. Tech. spr. Webglobe. Dostupné tiež z: <https://www.webglobe.sk/blog/html-zaklady>.
5. *CSS*. Coderama. Dostupné tiež z: <https://coderama.com/slovník/css>.
6. BITTNER, Honza. *Úvod do CSS preprocesora Sass*. ITnetwork. Dostupné tiež z: <https://www.itnetwork.sk/html-css/webove-portfolio/tutorial-moderne-webove-portfolio-sass>.
7. *JavaScript*. ITnetwork. Dostupné tiež z: <https://www.itnetwork.sk/javascript>.
8. KVAPIL, Jiří. *Úvod do TypeScriptu*. 2022. Tech. spr. Webglobe. Dostupné tiež z: <https://www.itnetwork.sk/javascript/typescript/uvod-do-typescriptu>.
9. *MathJax*. MathJax. Dostupné tiež z: <https://docs.mathjax.org/en/latest/basic/mathjax.html>.
10. *Backend*. Coderama. Dostupné tiež z: <https://coderama.com/slovník/backend>.
11. HOSSAIN, Monsur. *CORS in Action: Creating and consuming cross-origin apis*. Simon a Schuster, 2014.
12. SYED, Basarat. *Beginning Node.js*. Apress, 2014.
13. HAHN, Evan. *Express in Action: Writing, building, and testing Node.js applications*. Simon a Schuster, 2016.
14. MIKOVÁ, Tereza. *Aký je rozdiel medzi UX a UI, a prečo potrebujete na super web oboje*. 2024. Tech. spr. Webglobe. Dostupné tiež z: <https://www.webglobe.sk/blog/rozdiel-medzi-ux-a-ui>.
15. *Figma*. Coderama. Dostupné tiež z: <https://coderama.com/slovník/figma>.
16. *Angular Material*. Material-UI. Dostupné tiež z: <https://material.angular.io/>.

17. MÁCA, Jindřich. *Úvod do Angular frameworku*. itnetwork. Tech. spr. Dostupné tiež z: <https://www.itnetwork.sk/javascript/angular/zaklady/uvod-do-angular-frameworku>.
18. DOUŠKOVÁ, Alena, KARIKOVÁ, Soňa a BLAŽÍČEK, Tomáš. Gamifikácia–špecifikum multimediálneho vzdelávania slovenských učiteľ'ov v zahraničí. *Vzdělávání dospělých 2021*. 2022, s. 110.