

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor: 18. Informatika

Informační systém pro efektivní správu projektové soutěže AMAVET

Vilém Cerman

Pardubice 2024

INFORMAČNÍ SYSTÉM PRO EFEKTIVNÍ
SPRÁVU PROJEKTOVÉ SOUTĚŽE AMAVET

INFORMATION SYSTEM FOR EFFECTIVE
ADMINISTRATION OF PROJECT COMPETITIONS OF
AMAVET

AUTOR Vilém Cerman

ŠKOLA DELTA - Střední škola informatiky
a ekonomie, s.r.o.

KRAJ Pardubický

ŠKOLITEL Ing. Jiří Formánek

OBOR 18. Informatika

Prohlášení

Prohlašuji, že svou práci na téma *Informační systém pro efektivní správu projektové soutěže AMAVET* jsem vypracoval/a samostatně pod vedením Ing. Jiřího Formánka a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Dále prohlašuji, že tištěná i elektronická verze práce SOČ jsou shodné a nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a změně některých zákonů (autorský zákon) v platném znění.

Pardubice dne: _____

Vilém Cerman

Vilém Cerman

Poděkování

Chtěl bych poděkovat Ing. Jiřímu Formánkovi za odborné vedení práce, komunikaci s organizátory soutěže a celkový nápad na projekt. Dále děkuji Bc. Luboši Zapotočnému za správu produkčního prostředí a konzultace volby technologií a svému spolužákovi Tomáši Diblíkovi za testování bezpečnosti aplikace.

Abstrakt

Tématem práce je vývoj informačního systému pro projektové soutěže pořádané Asociací pro mládež, vědu a techniku AMAVET. V dokumentu je popsán průběh soutěží a úkony spojené s jejich organizací, slabiny původního řešení a krátký nástin obvyklých řešení podobných soutěží. Hlavní část práce je věnována návrhu a realizaci nového řešení. V závěru jsou zmíněny výhody nového řešení oproti starému a naznačen možný směr dalšího vývoje a zobecnění systému.

Klíčová slova

Informační systém pro soutěže, registrace, administrace, AMAVET, Next.js, relační databáze

Abstract

Subject of project is development of information system for project competitions held by Association for youth, science and technology AMAVET. In this document, I described competition itself and tasks connected with its organization, disadvantages of existing solution and short outline of common solutions of similar competitions. Main part of document is dedicated to designing and realization of new solution. In the conclusion I mention advantages of new solution against the old one and possibilities of future development.

Keywords

Information system, registration, administration, AMAVET, Next.js, relational databases

Obsah

Úvod	9
0.1 Kdo je AMAVET	9
0.2 Struktura soutěží	9
0.3 Průběh soutěže	10
1 Administrace soutěže - výchozí stav	11
1.1 Personální zajištění administrace soutěží	11
1.2 Roztříštěnost dat	11
1.3 Registrace soutěžících	12
1.4 Práce s daty soutěžních projektů	12
1.5 Hodnocení projektů	12
1.6 Tiskové výstupy	13
1.7 GDPR	15
1.8 Shrnutí nejpalčivějších problémů výchozího stavu	16
2 Obvyklá řešení	17
2.1 Microtela	17
2.2 SOČ - Středoškolská odborná činnost	17
2.3 ISEF - International Science and Engineering Fair	18
3 Tvorba vlastního řešení	19
3.1 Url adresa a přístupové údaje	19
3.2 Vymezení rozsahu práce	19
3.3 Volba technologií	21
3.3.1 Next.js	21
3.3.2 Databáze	21
3.3.3 Knihovny	21
3.4 Volba hostingu	22
3.4.1 Testovací prostředí	22
3.4.2 Produkční prostředí	23
3.5 Technologické výzvy	23
3.6 Backend	24
3.6.1 Autentizace	24
3.6.2 Spojení s databází	28

3.7	Frontend	29
3.7.1	Seznam projektů	29
3.7.2	Detail projektu	30
3.7.3	Registrační formulář	30
3.7.4	Hodnotitelská část aplikace	30
3.7.5	Administrace soutěže	31
3.7.6	Tiskové výstupy	32

Závěr

Literatura

Seznam obrázků

Úvod

Pro lepší pochopení celého projektu jsem se úvod rozhodl věnovat představení spolku AMAVET a soutěží, které pořádá.

0.1 Kdo je AMAVET

Asociace pro mládež, vědu a techniku, z.s. je spolek dětí, mládeže a dospělých, v jehož zájmu je vytváření a rozvíjení zájmu dětí a mládeže o vědu a techniku v České republice. Mezi jeho hlavní aktivity patří provozování zájmových vědeckotechnických center a soutěží, podněcující zájem o vědu a techniku u dětí a mládeže.[1]

Asociace pořádá množství vědeckých akcí, mezi které patří i **projektové soutěže**, mimo jiné Expo Science AMAVET a Festival vědy a techniky, dále jen FVT. Cílem těchto soutěží je podněcovat co nejvíce starších žáků základních škol a středoškoláků prostřednictvím řešení konkrétních vědeckých a technických projektů k odhalování a rozvíjení jejich tvůrčích schopností a schopnosti svoji práci prezentovat odborné i široké veřejnosti.[2]

0.2 Struktura soutěží

Amavet pořádá několik soutěží, zaměřených na konkrétní geografické oblasti, s názvem Festival vědy a techniky. Nejstarší a dosud největší z nich je Festival vědy a techniky v Pardubickém kraji (FVTP), čítající dvě okresní kola, v Pardubicích pro okresy Chrudim a Pardubice a v Lanškrouně pro okresy Svitavy a Ústí nad Orlicí. Nejlepší projekty z obou okresních kol postupují do krajského kola v Pardubicích.

Z dalších krajů má aktuálně svoji vlastní soutěž ještě kraj Královéhradecký (Festivaly vědy a techniky pro kraj Královéhradecký) a v minulosti míval i Liberecký, kterou se však po Covidu již nepodařilo obnovit. Královéhradecký i Liberecký kraj měly vždy pouze krajská kola. Projekty z ostatních regionů je možné hlásit do tzv. regionálního kola Expo Science AMAVET, které se obvykle koná v Brně.

Soutěže jsou dále rozděleny na dvě věkové kategorie: Junior, pro žáky páté až deváté třídy základních škol a odpovídajících ročníků víceletých gymnázií a Středoškolák, pro žáky prvního až čtvrtého ročníku škol středních.

Autoři vítězných projektů z kategorie Junior mají jako odměnu týdenní vzdělávací zájezd do Francie. Autoři vítězných projektů z krajských a regionálních kol v kategorii Středoškolák postupují do celostátního kola Expo Science AMAVET konajícího se na půdě Akademie věd ČR v Praze.

Z celostátního kola Expo Science AMAVET bylo historicky možné postoupit do několika mezinárodních soutěží (Soutěž Vernadského v Moskvě, Beijing Youth

Science Creation Competition, International Sustainable World Energy, Engineering, and Environment Project ISWEEEP v USA a International Science and Engineering Fair ISEF v USA). V současné době zůstala zachována již pouze možnost postupu na ISEF.[3]

0.3 Průběh soutěže

Autor/autoři přihlašují svůj projekt do základního kola soutěže podle regionu své školy. Základním kolem může být buď kolo okresní, krajské nebo i regionální - viz. Struktura soutěží. Každý projekt může mít maximálně tři autory.

Na přihlášce autoři uvádí kromě kontaktních údajů také základní informace o projektu: cíl projektu, popis projektu, vlastní přínos řešitele (řešitelů) a závěry. Veškeré údaje tvoří tzv. abstrakt, který má sloužit k předběžnému seznámení porotců s projektem.

Na začátku soutěže po zběžném seznámení s projekty si každý porotce vybere, které projekty bude hodnotit. Pro hladký průběh kola je ideální, aby každý projekt hodnotili aspoň tři porotci. Následně porotce obejde stanoviště jednotlivých projektů, kde soutěžící svůj projekt představí. Porotce pak v další diskusi s autory ověří kvality projektu případně širší vzhled do problematiky autorů.

Po skončení hodnocení se celá porota sejde, aby na základě zjištěných informací rozhodla o postupu jednotlivých projektů do dalších kol soutěže, případně o výsledném pořadí.

1 Administrace soutěže - výchozí stav

Ačkoliv následující kapitoly budou vyznívat možná až příliš kriticky, je třeba říct, že základní princip soutěží AMAVET dlouhodobě funguje. **Soutěže skutečně motivují a podporují žáky základních a středních škol k samostatné práci na svých vědeckých a technických projektech.**

Vytvořit strukturu soutěží v takovém rozsahu a udržovat ji po tak dlouhou dobu dalo jistě neuvěřitelné množství práce. Všechny kritické komentáře jsou vedeny pouze upřímnou snahou o zlepšení původního stavu.

1.1 Personální zajištění administrace soutěží

Věcnou administraci všech soutěží zajišťuje jediná osoba s relativně omezenou schopností ovládat informační technologie. Zvládá základní práci s Wordem a Excelem, ale již není schopna jakékoliv automatizace. Nezvládne používat (natož samostatně vytvořit) hromadnou korespondenci, přepsat .pdf dokument apod. Vytvoření a vytisknutí diplomů pro 150 soutěžících byl opravdu velký problém.

Základní snahou tedy bylo, správu všech soutěží co nejvíce sjednotit, zautomatizovat a ovládání udělat co nejsnazší.

1.2 Roztříštěnost dat

Data k soutěži byla evidována různým způsobem na různých místech.

- Primární data o projektech a autorech byla evidována v Google Tabulkách,
- data o jednotlivých kolech soutěží (datum, místo) byla v e-mailových pozvánkách,
- vlastní text e-mailové pozvánky byl evidován v Outlooku organizátorů AMAVET,
- seznamy a kontaktní informace na porotce byly pravděpodobně v lokálních souborech na osobním počítači organizátorů AMAVET,
- přiřazení porotců k projektům bylo pouze v papírových poznámkách jednotlivých porotců,
- dílčí hodnocení projektů bylo opět pouze v papírových poznámkách jednotlivých porotců,
- výsledné hodnocení projektů bylo v poznámkách organizátorů AMAVET, na diplomech a ve výsledkové listině.

Tato roztříštěnost znemožňovala jakoukoliv komplexnější automatizaci a přinášela velké množství provozních problémů.

1.3 Registrace soutěžících

Registrace projektů do soutěže probíhala prostřednictvím formulářů Google Forms, umístěném na webových stránkách AMAVETu. Pořízená data byla automaticky uchovávána propojením s Google Tabulkami. Každý rok se data z aktuálních tabulek zkopírovala do Excelu pro další manuální práci a data v Google Tabulkách se "vyčistila".

Jak jednotlivé soutěže historicky přibývaly, tak se i struktura dat pro každou soutěž mírně lišila, což značně komplikovalo jejich další zpracování.

1.4 Práce s daty soutěžních projektů

Excelovské tabulky byly vytvářeny pro každé kolo soutěže samostatně. Perličkami pro dokreslení pracnosti může být, že rozřazování projektů do okresních kol v Pardubickém kraji probíhalo manuálně podle obce školy. Manuálně také probíhalo přiřazování pohlaví všem autorům projektů - kvůli správnému pravopisu textů na diplomech.

Z primárních dat byly manuálně vytvářeny jejich kopie a z nich pak různé tiskové výstupy - především vizitky a diplomy. Vizitka byla připravována pro každého soutěžícího, takže u vícečlenných týmů bylo třeba manuálně extrahovat jednotlivé soutěžící a každému vytvořit samostatný záznam. S každým projektem, který postoupil přes okresní a krajské kolo až do národního, tyto úkony proběhly 3x.

Veškeré zpracování bylo extrémně pracné, časově náročné a náchylné k manuálním chybám. Excelovské tabulky s přihlášenými projekty byly uchovávány lokálně a nebyly nijak zálohovány.

1.5 Hodnocení projektů

Porotcům byl e-mailem rozeslán seznam všech projektů společně s tzv. abstrakty (základní informací o projektu) s přibližně týdenním předstihem, aby se s projekty předem seznámili. Přiřazení konkrétních projektů ke konkrétním porotcům nebylo předem určeno a každému porotci přišel soubor např. se 100 projekty. Naprostá většina porotců tak tento e-mail ignorovala.

Celá porota se sešla cca 1 hodinu před oficiálním zahájením soutěže. Každý porotce dostal seznam všech projektů, které byly očíslované. Předseda hodnotitelské komise četl popořadě seznam projektů a porotci se podle názvů a vědních oborů k jednotlivým oborům "hlásili". Pokud se k projektu slovně přihlásili aspoň 3 porotci, bylo přiřazení považováno za vyřízené. **Informace o přiřazení porotců k projektům byla pouze v poznámkách jednotlivých porotců a nebyla ni-**

kde centrálně evidována. Stávalo se, že některý porotce se přihlásil k velkému množství projektů a nestihl je všechny ohodnotit. Občas nastala i situace, kdy některý projekt nepřišel ohodnotit žádný porotce.

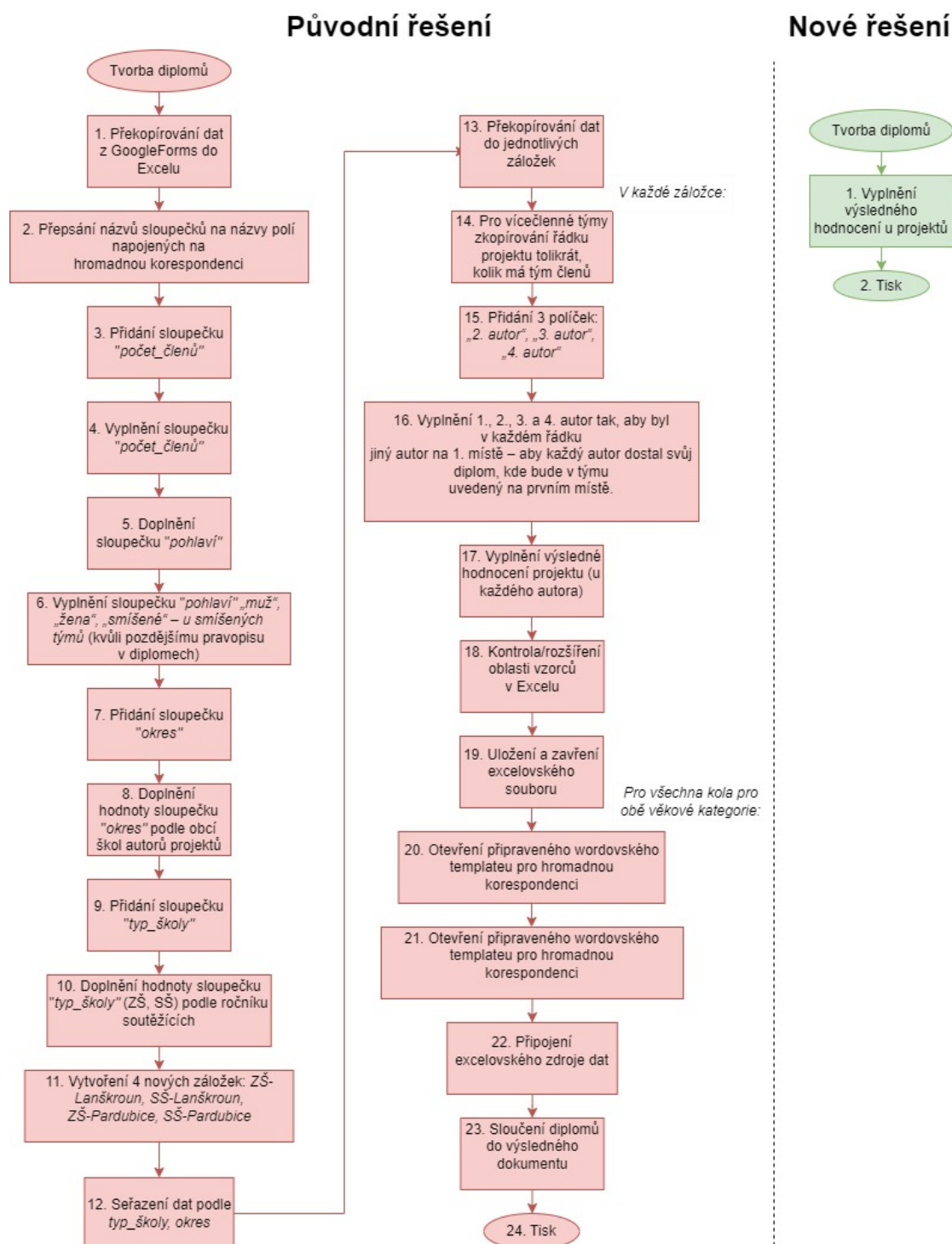
Po oficiálním zahájení začali porotci hodnotit projekty. Prohlédli si prezentační poster a rozhovorem se soutěžícím se snažili posoudit kvalitu projektu. Do svých papírových seznamů si mohli psát poznámky k hodnocení.

Po cca 2-3 hodinách hodnocení se znovu sešla hodnotitelská komise. Její předseda opět četl seznam projektů a jednotliví hodnotitelé hlásili svá hodnocení. O každém projektu se jednalo až do dosažení konsenzu, zda projekt "uspěl" a postupuje do dalšího kola, nebo "neuspěl". Výsledek si poznamenávala pomocnice předsedy hodnotitelské komise na papírový seznam projektů. (Pokud udělala chybu, neexistoval žádný mechanismus nápravy.) Jednotlivá dílčí hodnocení projektů byla evidována pouze v osobních poznámkách porotců. Centrální evidence neexistovala. Bez přítomnosti jednotlivých hodnotitelů nebylo možné hodnocení revidovat. V případě, že některý soutěžící rozporoval své hodnocení, nebylo možné se odkázat na dílčí hodnocení konkrétních porotců a byl odkázán pouze na vágní "Hodnotitelská komise rozhodla ...".

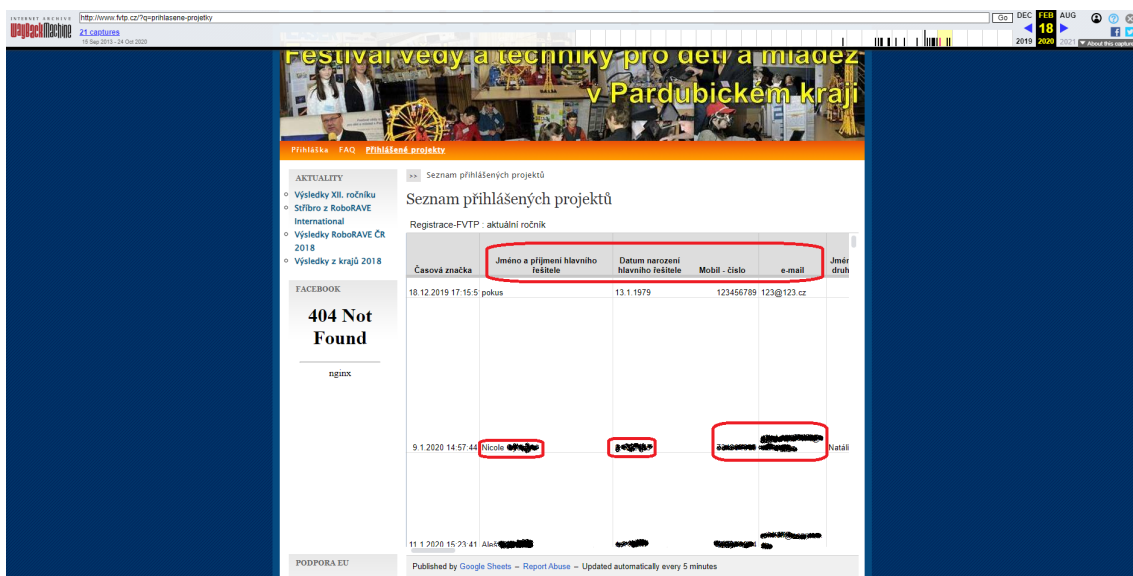
Teprve v okamžiku konečného hodnocení se občas zjistilo, že některé projekty nehodnotil žádný porotce a šlo se tyto projekty "dohodnotit".

1.6 Tiskové výstupy

Organizace takových soutěží se neobejde bez množství tiskových výstupů, od prostých seznamů, prezenčních listin, seznamu e-mailových adres a výsledkových listin až po výstupy vyžadující precizní pozicování - vizitky a diplomy. Ve výchozím stavu byly tiskové výstupy vytvářeny manuálně s minimálními prvky automatizace. Tvorba výstupů zabírala velké množství času, byla nesmírně pracná a náchylná na chyby. To se projevovalo obzvlášť při tvorbě diplomů, které se mohly připravovat až po skončení celkového hodnocení a které se vždy připravovaly ve velké časové tísní. I když na tisk diplomů měli organizátoři soutěže připravené šablony pro hromadnou korespondenci, neuměli je využívat. Příprava datových podkladů pro hromadnou korespondenci si zabírala pro jedno soutěžní kolo 4-5 hodin.



Obrázek 1.1: Původní řešení tisku diplomů



Obrázek 1.2: Porušení GDPR původního systému

1.7 GDPR

Všechny soutěže AMAVETu vznikly historicky ještě před platností GDPR. Jejich organizátor toto nařízení zcela ignoroval a tak ještě v minulém roce bylo možné na stránkách amavet.cz i fvtp.cz nalézt vložený iframe, který vedl na Google tabulky s kompletními daty přihlášených projektů. Zcela volně dostupné byly informace: Jméno a příjmení autora, datum narození, číslo mobilního telefonu, e-mailová adresa, název a adresa školy, ročník,... (viz. obr.1.2)

Podle serveru edevizy.cz činila průměrná výše pokuty v ČR v roce 2023 za porušení GDPR 2,5 mil. Kč a Úřad pro ochranu osobních údajů řešil 5 547 stížností na porušení.[4]

1.8 Shrnutí nejpálčivějších problémů výchozího stavu

- Velká roztržitost uchovávaných dat - různé formy na různých místech.
- Chyběla jakákoliv automatizace tvorby tiskových výstupů.
- Neexistovala žádná možnost zpětně dohledat podklady dílčích hodnocení porotců (ani kteří porotci konkrétní projekt hodnotili).
- Zveřejňováním osobních informací o soutěžících i další manipulací s daty docházelo k vážnému porušování nařízení GDPR.

2 Obvyklá řešení

Soutěžemi podobnými Expo Science AMAVET jsou například: lokální soutěž Microtela v Lanškrouně, Středoškolská odborná činnost a v zahraničí pak například International Science and Engineering Fair v USA. Chtěl jsem udělat porovnání jejich informačních systémů a stupně automatizace jejich provozu. Bohužel mám však jako běžný uživatel přístup pouze k jejich veřejné části a z té toho nezjistím mnoho.

2.1 Microtela

Lokální soutěž v Lanškrouně, která se jednou ze svých částí nechala hojně inspirovat AMAVETem. Letos proběhne 4. ročník soutěže. Obsahuje poměrně velké množství kategorií a různých "podsoutěží". Je určena pro děti ve věku od mateřských školek až po druhý stupeň ZŠ. Různé kategorie mají různá zaměření, ale celkové zaměření soutěže je technika, robotika, programování. Má jednoduchý web www.microtela.cz, který obsahuje základní data o soutěži (včetně archivu) a možnost přihlášení se. Přihlašování je připraveno přes GoogleForms - což odpovídá "rannému věku" soutěže. Veškeré správa soutěžících včetně tisku diplomů, vytváření výsledkových listin apod. bude pravděpodobně probíhat manuálně (i podle rychlosti reakce pana starosty se zájmem o spolupráci, když viděl informační systém pro AMAVET). Soutěž bude pravděpodobně mít hodně podobné požadavky na automatizaci jako AMAVET.

2.2 SOČ - Středoškolská odborná činnost

SOČ je nejstarší a nejprestižnější projektová soutěž středoškoláků v ČR. Je rozdělena do 18ti oborů a geograficky hierarchicky uspořádána do postupových kol (okresní, krajská, celostátní). Letos proběhne již 46. ročník soutěže.

Pro přihlášení projektu do soutěže využívá vlastní informační systém. Heslo je možné získat na adrese <https://prihlaska.soc.cz> - automaticky přijde na zadaný e-mail. Po přihlášení do systému soutěžící vyplní své adresní údaje, údaje o práci, přiloží odkaz na video a samotnou dokumentaci své práce.

Samotný systém svůj účel splňuje, nicméně je na jeho vzhledu vidět, že byl vytvořen před delší dobou. Proto by se mu hodilo aspoň mírné "omlazení".

Z hlediska UX i konzistence dat by bylo vhodné, aby se při vyplňování názvu školy začaly nabízet školy odpovídající vyplňovanému textu a při výběru patřičné školy, by se doplnily její veškeré adresní údaje. (Tak to funguje nyní při zadávání do systému AMAVET.)

Je otázkou, co by se mělo správně stát, když dva studenti z jedné školy vyplní jako garanta školního kola dva různé učitele. - V návrhu datového modelu IS jsou pravděpodobně chyby, které mohou vést k nekonzistenci dat.

Systém neposkytuje nikde seznam soutěžních prací s jejich hodnocením - úspěchy. Pokud by například bylo možné filtrem nastavit takovou url adresu, která by toto umožňovala za projekty jedné školy - tak by toho jistě každá škola ráda využila ke své propagaci. To by zpětně vedlo k propagaci SOČ.

Nicméně je třeba SOČ pochválit, že shromažďuje veškerá data o projektech, soutěžících a jejich školách a data o struktuře soutěže (jednotlivých hierarchicky uspořádaných kolech) v jedné relační databázi. Zda systém zahrnuje i správu porotců, přiřazování projektů porotcům, zapisování dílčích hodnocení, celková hodnocení, tisky diplomů a například kopii celé struktury do dalšího roku, není možné z mého pohledu posoudit. Každopádně je automatizace v SOČ na výrazně lepší úrovni, než byla před rokem v Expo Science AMAVET.

2.3 ISEF - International Science and Engineering Fair

ISEF je nejstarší a největší mezinárodní soutěží středoškolských projektů na světě. Je to jedna ze soutěží, kterou pořádá mezinárodní spolek Society for Science. Disponuje rozsáhlým webem www.societyforscience.org/isef/, který kromě standardních informací o soutěži pokrývá uvolňování tiskových zpráv pro média, možnost přihlášení dobrovolníků, porotců, firem/institucí, které chtějí udílet zvláštní ceny a podobně. Zkrátka soutěž pokrývají zcela komplexně. Grafika webu navíc působí příjemně a moderně.

Systém obsahuje všechny soutěžní projekty, včetně grafického znázornění získaných ocenění. Bohužel se odkaz na konkrétní projekt nepropisuje do url adresy, aby školy mohly využít detailní info o projektu na oficiálních stránkách ke své propagaci.

Podle komplexnosti pokrytí informací a zobrazování výsledného hodnocení u projektů lze předpokládat, že za prezentační částí stojí robustní komplexní informační systém.

V ISEF každoročně soutěží cca 1300 projektů z celého světa, jsou zde řádově stovky hodnotitelů a institucí udílející zvláštní ceny. Soutěž takového rozsahu bez opravdu robustního informačního systému je prakticky nemyslitelná. S více než 70-tiletou tradicí měla dostatek času a síly si jej vytvořit.

3 Tvorba vlastního řešení

Bylo zřejmé, že původní řešení je zcela nevyhovující a že je pouze otázkou času, kdy se systém obsluhy soutěží zhroutí. Rozhodl jsem se vytvořit informační systém, kde budou veškerá pořizovaná data ukládána v relační databázi a veškeré zpracování výstupů bude automatizováno. Jednou naprogramovaný a odladěný algoritmus nebude generovat "lidské" chyby, jeho provoz bude řádově rychlejší a ovládání jednodušší.

3.1 Url adresa a přístupové údaje

Pro testování projektu pro porotce zde uvádím adresu produkční i testovací verze systému a pro testovací verzi i dočasné přihlašovací údaje administrátora. Tyto údaje zneplatním k 30.6.2024. Zároveň testovací login je neexistující e-mailová adresa, proto nefungují funkce jako zapomenuté heslo. Pro plné otestování autentizace a autorizace si prosím vytvořte vlastní porotcovský účet, který si můžete pomocí adminského účtu přiřadit k dostupným kolům.

Pro testování projektu pro účely SOČ jsem použil produkční data, ze kterých jsem nahradil příjmení autorů jejich křestním jménem a kontaktní informace smyšlenou konstantní hodnotou.

Popis	Hodnota
produkční verze	https://amavet.delta-skola.cz
testovací verze	https://amavet.rostiapp.cz
login admina	porotce@example.com
heslo admina	HesloProPorotce2024

3.2 Vymezení rozsahu práce

Původní zadání projektu bylo velice vágní: **"Ze systému musí stávající obsluha snadno tisknout abstrakty projektů, vizitky a diplomy."** To však bylo v původním systému zcela nezrealizovatelné.

Jednoznačně jako první úkol bylo třeba dostat všechna data na jedno úložiště. Potom bylo nezbytné si vymežit rozsah projektu, jakou funkcionalitu budu řešit v první verzi a jakou bych chtěl vyřešit v budoucnu (abych si třeba budoucí řešení neztížil nebo zcela neznemožnil).

V první verzi jsem se rozhodl řešit tyto oblasti:

- definice hierarchické struktury kol soutěží (byť její uživatelská administrace

může být odložena),

- registrace projektu uživatelem do soutěže,
- přiřazení identifikačních čísel projektům v rámci kola podle vědních oborů,
- registrace porotců a jejich přiřazení k jednotlivým kolům soutěží,
- přiřazení porotců k projektům pro dílčí hodnocení,
- prohlížení abstraktu projektů porotci, aby se odstranila nutnost tisku abstraktů,
- dílčí hodnocení projektů porotci,
- výsledné hodnocení projektů v rámci kola,
- tisk vizitek (soutěžících, porotců),
- tisk diplomů soutěžícím,
- vytváření uživatelského účtu porotce, možnost přihlášení, možnost obnovy hesla,...,
- umožnění prohlížení seznamu projektů nepřihlášeným uživatelem, možnost řazení a filtrování (při dodržení zásad GDPR),
- promítání filtrů do url adresy tak, aby školy mohly adresu "svých" projektů umisťovat na vlastní webové stránky školy

Při předávání řešení se vyskytly požadavky na novou funkcionalitu. Některé byly dodělány (tisková sestava - prezenční listina soutěžících, prezenční listina porotců, výsledková listina, seznamy projektů,...), jiné byly zaregistrovány pro následující verzi systému.

Do budoucna bude dále nezbytné vyřešit:

- kompletní uživatelskou administraci jednotlivých kol soutěže včetně nastavování hierarchie postupů (nyní lze nastavovat pouze přes databázové rozhraní)
- automatizovat postup do vyšších kol soutěží (nyní lze spustit skriptem z databáze),
- kompletní uživatelskou administraci textů na diplomech (nyní lze pouze přes databázové rozhraní),

- uživatelské rozhraní pro zkopírování struktury soutěží do nového roku včetně přiřazených porotců (datová struktura funkcionalitu umožňuje, zbývá dopsat uloženou proceduru a napojit na uživatelské rozhraní),
- kompletní uživatelskou administraci pro automatizovanou e-mailovou komunikaci s porotci a účastníky soutěže (technologie je zvládnutá, datová struktura funkcionalitu umožňuje, zbývá doprogramovat)

3.3 Volba technologií

Již při vymezení rozsahu práce bylo zřejmé, že zhotovení takového systému se všemi požadovanými funkcemi bude náročné především časově. Snažil jsem se proto zvolit snadno rozšiřitelnou modulární architekturu a široce rozšířené technologie umožňující rychlý vývoj.

V následujících kapitolách je popsán zvolený technologický pool. U každé technologie jsou uvedeny i její alternativy a má volba je zdůvodněna.

3.3.1 Next.js

Z výše popsaných důvodů byl zvolen pro práci framework Next.js od společnosti Vercel[5]. Tento framework je silný nástroj pro tvorbu webových aplikací pro svoji jednoduchou škálovatelnost díky React komponentám¹, vlastnímu Route Handleru² a jednoduchému využití knihoven třetích stran (dále jen knihovny).

Při volbě technologie jsem zvažoval jako alternativu PHP[6], už jen proto, že s ním mám zkušenosti ze školy a už jsem v něm podobnou aplikaci vytvářel. PHP je na trhu poměrně dlouho, je dobře zdokumentované a na internetu jsou dohledatelná řešení většiny problémů. Next.js je razantně nastupující relativně nová technologie a o programátory, kteří ji ovládají, je na pracovním trhu velký zájem. To byl také jeden z hlavních důvodů, proč jsem zvolil pro řešení projektu právě ji.

3.3.2 Databáze

Jako databáze byla zvolena relační databáze MySQL, jejíž instalace poskytuje i jednoduchý nástroj Adminer pro snadnou správu dat. Právě ten, mi umožňuje manipulaci s daty v případě, že by byla některá část aplikace nefunkční. Pro snadný návrh datového modelu byl použit nástroj MySQL Workbench.

3.3.3 Knihovny

Systém využívá několik knihoven. Knihovny obecně dovolují použít programátorům kód, který již připravil někdo jiný a který řeší požadovanou skupinu problémů.

¹React komponenty - dovolují používání stejného kódu na více místech, komponenta může být jednoduché tlačítko, nebo část stránky

²Route Handler - stará se o navigaci mezi jednotlivými stránkami webové aplikace

Jejich použitím si programátoři šetří práci. Některá řešení jsou však příliš konkrétní nebo naopak obecná a jejich implementace často vyžaduje další uzpůsobení nebo i přepsání požadované funkcionality (viz. kapitola NextAuth.js). I přesto mi implementace právě NextAuth.js ušetřila velké množství práce a času.

MySQL2

Aplikace téměř neustále komunikuje s databází a k tomu jsem zvolil knihovnu MySQL2[7]. Knihovna řeší kromě samotného spojení s databází i takzvané připravení dotazů (z angl. "prepare"). Pod to spadá cachování, validování dat, převádění typů mezi MySQL a JavaScriptem a ochrana před útoky typu SQL Injection³.

Bcrypt

Aplikace podporuje přihlášení pomocí hesla, které je uloženo v databázi. Z bezpečnostních důvodů ale nelze v databázi uchovávat samotné heslo v čitelné podobě a proto je potřeba ho zašifrovat. Na šifrování je využita Node.js knihovna Bcrypt[8], využívající stejnojmenného algoritmu pro šifrování. [9]

Tento algoritmus pomocí matematické rovnice přemění text na nečitelnou a na první pohled náhodnou sekvenci znaků. Pro lepší bezpečnost je k původnímu textu přimíchána takzvaná kryptografická sůl. Tímto se předchází útokům pomocí rainbow table.

NextAuth.js

Součástí systému je autentizace a autorizace uživatelů, umožňující rozdělení do několika rolí, z nichž každá má jiné pravomoce. V rámci autorizace je potřeba zabezpečit některé stránky a endpointy jak na klientovi tak na serveru. K tomu byla vybrána knihovna NextAuth.js, momentálně nejpoužívanější řešení zabezpečení k frameworku Next.js. Knihovna nabízí obecné řešení zabezpečení aplikace, které se však soustředí na autentizaci pomocí třetích stran, například přes účet Google nebo GitHub.

3.4 Volba hostingu

3.4.1 Testovací prostředí

Pro snazší koordinaci vývoje projektu mezi autorem a vedoucím práce vyžadoval vedoucí veřejně přístupné testovací prostředí (ne pouze lokální vývoj). Nasazování nových verzí na testovací prostředí několikrát prokázalo, že když něco funguje

³SQL Injection je typ útoku na databázi kde útočník vloží do dotazu místo hodnoty (např. čísla nebo hledaného slova) svůj kód, ve kterém příkaz ukončí a za ním vykoná svůj

na lokálním prostředí, tak to ještě neznamená, že to bude fungovat na veřejném hostingu. Tento postup odhalil několikrát chybu ještě před nasazováním na produkční prostředí.

Volba veřejného testovacího prostředí ukázala své opodstatnění i ve chvíli, kdy již musela běžet registrace projektů na produkčním prostředí, ale další funkcionality systému se stále vyvíjela.

Pro testování jsem si zvolil hosting od společnosti Roští.cz. Dle mého průkumu je to jeden z nejlepších hostingů technologie Next.js v České republice, hlavně díky velké míře automatizace a snadno dostupné podpoře. Na tomto prostředí je také přístupná testovací verze pro porotce, viz. sekce Url adresa a přístupové údaje.

Zvažovanou alternativou k zvolenému testovacímu hostingu byl Exon.io[10]. Je to další hosting v České republice s podporou Next.js. Nabízí špičkovou podporu a dokonce lepší specifikace a výpočetní výkon než na Roští.cz. Na svou aplikaci jsem ale neměl tak velké výkonostní nároky a Exon.io tarify se právě kvůli nim pohybují v řádově vyšších cenových hladinách. S nasazováním Next.js aplikací jsem neměl dosud žádné zkušenosti a Roští.cz nabízí poměrně rozsáhlou dokumentaci prostředí a všech technologií.

3.4.2 Produkční prostředí

Produkční prostředí pro chod aplikace zajišťovala škola. Obsahuje i další aplikace spravované školou, proto do něj nemám přímý přístup. Nasazování nových verzí probíhá automaticky prostřednictvím GitHub workflow, který si naklonuje mnou zpřístupněný GitHub repozitář, aplikaci sestaví spolu s ostatními technologiemi do jednoho Docker image[11], který je poté nasazen na produkční prostředí.

3.5 Technologické výzvy

Z technologického hlediska bylo nezbytné vyřešit následující úkoly:

- komunikaci s databází, tzv. EndPointy,
- zobrazování dat (základní vzhled prezentovaných dat, třídění, filtrování, responzivita,...),
- autentizace uživatelů,
- rozesílání e-mailů,
- generování .pdf

V následujících kapitolách je řešení jednotlivých výzev detailně popsáno.

3.6 Backend

Backend je logická část aplikace, která se stará především o práci s daty a jejich předání frontendové části aplikace. Komunikace mezi nimi probíhá přes rozhraní REST API. Toto rozhraní je veřejné a přístupné odkudkoliv, nejen z frontendové části aplikace. Zároveň ale v mnoha případech posílá citlivá data, musel sem proto implementovat ověření uživatelů. K tomuto jsem použil knihovnu NextAuth.js (viz. sekce NextAuth.js).

3.6.1 Autentizace

Možnost autentizace knihovna nabízí na catch-all cestě `/api/auth/[...nextauth]`. Catch-all cesta (překl. zachytí vše) je adresa, která zachytí všechny požadavky začínající definovanou cestou a zbytek adresy pak předává kódu jako parametr. Například catch-all cesta `/projekt/[...projektId]` zachytí požadavek na `/projekt/123` a kódu předá zbytek cesty v požadavku jako parametr (v tomto případě `projektId = 123`).

Tento endpoint nabízí několik služeb, spojených s autentizací a autorizací. Při parametru `/api/auth/csrf` endpoint vrací csrf token, kterým se poté klient prokazuje při dalších akcích. Parametr `/api/auth/providers` vrací seznam všech možností autentizace - auth-providerů. Moje aplikace podporuje dvě:

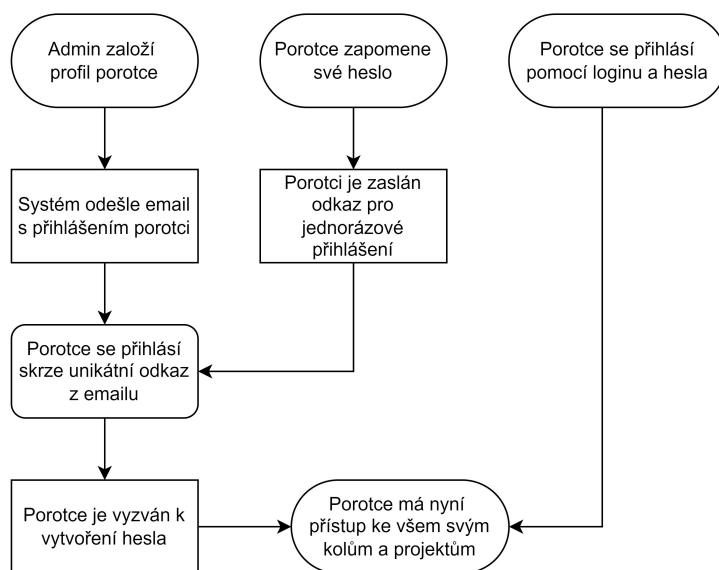
- Credentials provider - přihlášení pomocí uživatelského jména (e-mailové adresy)
- Email provider - použitý k registraci porotců pomocí emailových pozvánek

Jednotlivé způsoby přihlášení a registrace porotců a využití obou providerů je popsáno v diagramu viz. obrázek 3.1.

Pro samotné přihlášení uživatelským jménem a heslem je použita metoda POST `/api/auth/callback/credentials`. Ta vyžaduje při nejmenším tři parametry: uživatelské jméno (e-mailovou adresu), uživatelské heslo a csrf token, který klient obdrží na zmíněném endpointu. Na serveru se poté ověří, jestli klient odeslal platný csrf token, poté jestli uživatel s daným emailem vůbec existuje a nakonec se porovná obdržené heslo s hashem původního hesla z databáze. Pokud vše sedí, uživatel je přihlášen a je mu vytvořena session s JWT tokenem, kterým se nadále klient prokazuje, místo opakovaného přihlašování.

Vlastní uzpůsobení autentizace

Jak je již uvedeno výše, knihovna NextAuth.js je napsaná velmi obecně a soustředí se na metody ověření, které však nevyhovovaly účelům mé aplikace. Knihovna



Obrázek 3.1: Možnosti přihlášení porotců

je však modulární a umožňuje snadnou náhradu funkcí za vlastní kód.

Příkladem je databázový adaptér pro MySQL. Knihovna má dobře zdokumentováno, jaké tabulky, hodnoty a jejich datové typy je potřeba využívat. Podle toho jsem navrhl strukturu, která se jednoduše napojuje na zbytek mé databáze. Následně jsem doprogramoval adaptér, který propojoval knihovnu s databází. Ten se skládal z několika metod, které měly definovaný název a typ návratových hodnot. Úryvek kódu je popsán na obrázku 3.2.

Knihovna podporuje, ale přímo neimplementuje možnost registrace pomocí unikátního odkazu zasláném e-mailem. Knihovna je schopná odkaz vygenerovat a pomocí mého MySQL adaptéru uložit do databáze. Samotné odesílání odkazu pomocí knihovny nodemailer jsem doprogramoval. Knihovna nodemailer se připojuje na SMTP server Seznam.cz, který email odešle adresátovi z adresy `registrace-amavet@seznam.cz`.

Uživatelské role

V aplikaci se s jednou ze tří definovaných uživatelských rolí:

- Nepřihlášený uživatel - Tento uživatel má nejmenší oprávnění a může pouze přihlásit svůj projekt a zobrazit si již přihlášené projekty. Přihlášené projekty se také zobrazují bez jakýchkoliv osobních údajů, aby se zamezilo porušení GDPR. Veškerá ostatní funkcionalita je běžnému nepřihlášenému uživateli odepřena.

```

// Metoda obdrží vše kromě id uživatele, id je implementováno v databázi
async createUser (newUser: Omit<AdapterUser, "id">) {
  try {
    const exists = await query({ // Vypiši jestli uživatel s daným emailem již neexistuje
      query: 'SELECT * FROM 18uzivatel WHERE a18email = ?',
      values: [newUser.email]
    }) as User[]

    let user: AdapterUser

    if (exists.length > 0) { // Pokud existuje, vrátím existujícího uživatele
      user = User.ToAdapterUser(exists[0])
    } else {

      const result = await query({ // Jinak uživatele uložím do databáze
        query: 'INSERT INTO 18uzivatel (a18email, a18emailverified) VALUES (?, ?)',
        values: [newUser.email, newUser.emailVerified]
      }) as any

      user = {
        ...newUser,
        id: result.insertId,
      } as AdapterUser
    }

    return user
  } catch (err) { // Zachytím případné chyby a ty vypíšu pro možnost zpětného dohledání
    console.log('Failed to create user: ', err)
  }
},

```

Obrázek 3.2: Metoda vytváření uživatele

```

EmailProvider({
  server: {
    host: process.env.EMAIL_SERVER_HOST,
    port: process.env.EMAIL_SERVER_PORT,
    auth: {
      user: process.env.EMAIL_SERVER_USER,
      pass: process.env.EMAIL_SERVER_PASSWORD
    },
  },
  from: process.env.EMAIL_FROM,
  sendVerificationRequest: sendVerificationRequestSmt,
  maxAge: 60 * 60 * 24 * 7, // 7 days
})

```

Obrázek 3.3: Email provider

```

import {SignInEmailReact, SignInEmailHtml} from '@app/emails/sign-in-email'
import { createTransport } from "nodemailer"

export async function sendVerificationRequestSmtp(params:any) {
  const { identifier, url, provider } = params
  const { host } = new URL(url)

  // Vytvořím transport se SMTP serverem Seznam.cz pomocí knihovny nodemailer
  const transport = createTransport(provider.server)
  const result = await transport.sendMail({
    to: identifier,
    from: provider.from,
    subject: `Přihlášení ${host}`,
    text: text({ url, host }), // textová verze emailu
    html: SignInEmailHtml({ url, host }) // Použiji vlastní html šablonu emailu
  })

  // Zachytím možné chyby v odesílání emailů
  // Může být chyba v přenosu, neexistující adresát, ...
  const failed = result.rejected.concat(result.pending).filter(Boolean)

  // Chyby vypíšu pro zpětnou dohledatelnost
  if (failed.length) {
    console.log('Failed to send the verification Email:', failed)
  }
}

// Textová verze emailu
// - může se stát že adresát otevírá emaily v aplikaci, která nepodporuje html
function text({ url, host } : {url: string, host: string}) {
  return `Sign in to ${host}\n${url}\n\n`
}

```

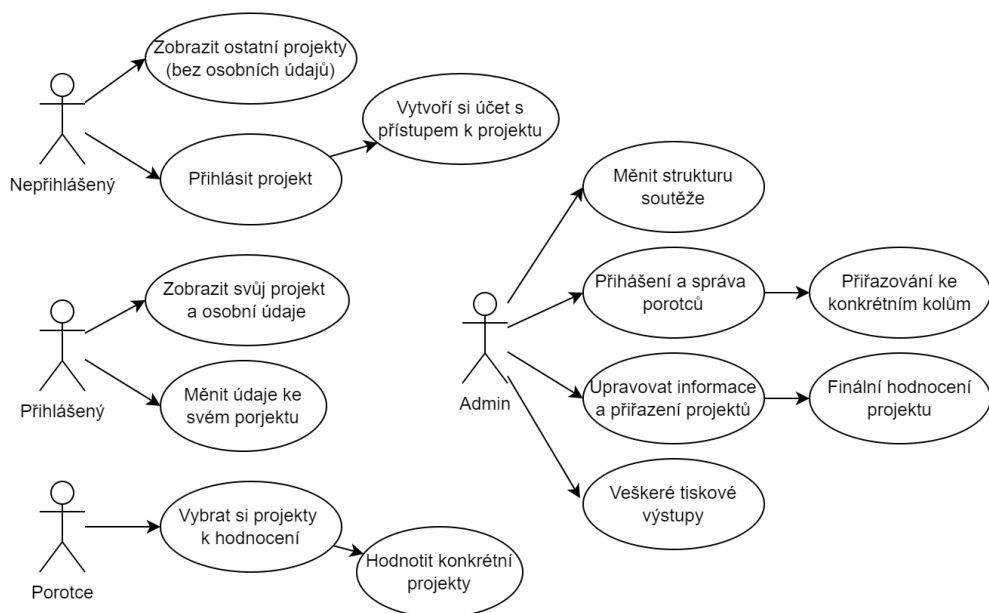
Obrázek 3.4: Odesílání přihlašovacích e-mailů

```

EMAIL_SERVER_USER='user'
EMAIL_SERVER_PASSWORD='heslo'
EMAIL_SERVER_HOST='smtp.seznam.cz'
EMAIL_SERVER_PORT='465'
EMAIL_FROM='registrace-amavet@seznam.cz'

```

Obrázek 3.5: ENV proměnné pro SMTP



Obrázek 3.6: Původní use case diagram

- Porotce - Přihlášeným uživatelům s rolí porotce je přístupná sekce Hodnocení (viz. kapitola Hodnotitelská část aplikace)
- Administrátor - Přihlášený uživatel s rolí administrátora, má největší oprávnění. V sekci Administrace může měnit různé parametry soutěží, zakládat hodnotitelské účty a další úkony popsané v kapitole Administrace soutěže

Kromě těchto rolí, byla v původním návrhu aplikace v plánu ještě jedna role: Přihlášený uživatel - Autor projektu. Po přihlášení projektu do soutěže se měl hlavnímu autorovi práce vytvořit uživatelský účet, spojený s jeho e-mailovou adresou. Pomocí tohoto přístupu by mohl během jisté ochranné lhůty měnit údaje uvedené v přihlášce, možné překlepy, faktické chyby a podobně. Nicméně mi v závislosti na rozsahu celého projektu nezbyl čas na implementaci role a veškeré její funkcionality. Tu mám v plánu implementovat v budoucnosti a v menším časovém presu, aby nedošlo k chybě v bezpečnosti aplikace.

3.6.2 Spojení s databází

Hlavním účelem aplikace je zobrazování, ukládání a práce s daty. Backend aplikace poskytuje endpoint pro každý typ dat zvlášť, tím rozdělují logické celky od sebe. Zároveň například vypisování seznamu projektů se v aplikaci objevuje mnohokrát, používá více způsobů filtrování a vypisuje občas další typy dat, přesto jsem se rozhodl veškerou funkcionalitu zachovat na jednom endpointu. Dovoluje mi to mít jednu modulární metodu a případné změny struktury dat řešit na jednom místě.

Obrázek 3.7: Ukládání filtrů v URL adrese

Pro komunikaci backendu s databází používám již zmíněnou knihovnu MySQL2 (viz. MySQL2). Knihovna je inicializována v jedné univerzální funkci, kterou používá celá aplikace. Na jednom místě je tak definované nastavení připojení a všechny údaje k připojení uložené v tzv. ENV proměnné⁴. Citlivé údaje jsou tak uloženy mimo veřejný repozitář a nelze je zjistit dekompilováním aplikace. Příklad ENV proměnné je na obrázku 3.5.

3.7 Frontend

Frontend je klientská část aplikace, která se zobrazuje běžnému uživateli. Ve své podstatě slouží k zobrazování dat, obdržенých z backendové části aplikace. Přesto jsem zde řešil technické problémy, jako je tisk diplomů a vizitek (viz. Generování PDF).

3.7.1 Seznam projektů

Úvodní stránka, kterou uživatel vidí při vstupu na stránky, je seznam projektů přihlášených do soutěže. Pro zobrazení projektů používám komponentu NextUI Table. Ta mi dovoluje dynamicky měnit její obsah, bez nutnosti renderovat celou stránku znovu. To je užitečné hlavně při filtrování projektů a řazení projektů. Projekty lze řadit podle několika kritérií: obor, kategorie, škola ze které se autoři hlásí, ročník soutěže a samotná kola, do kterých jsou projekty přihlášeny. Všechny tyto filtry jsou dynamicky získávány z databáze a nemusím tak stránku přepisovat pokaždé, když například přidá další obor. Zároveň aplikace poskytuje vyhledávání pomocí názvu projektu.

Všechno filtrování se ukládá do parametrů URL adresy, k tomu mě vedly dva důvody. Filtrování se zachová při navigování mezi stránkami pomocí tlačítek navigování v historii prohlížení. Tato historie uchovává pouze URL adresu stránky a kdyby filtrování bylo uloženo v proměnné v kódu, při každém navigování by se při kompilaci vyresetovalo. Druhým důvodem je možnost sdílení stránky i s filtrováním přes odkaz na stránku i s parametry (Obrázek 3.7), nebo třeba vložení pomocí IFrame na jinou webovou stránku.

⁴ENV proměnná (z angl. Environmental variable) slouží k ukládání údajů které jsou buď citlivé, nebo závisí na produkčním prostředí. Aplikace se kompiluje bez nich a můžeme je tak libovolně přidávat a měnit nezávisle na aplikaci

```
// ...
return(
  <div className="w-full lg:w-card-lg p-8 flex flex-col items-start gap-8 m-auto">
    { /* Komponentě buď předám samotná data */ }
    <ProjectDetail data={data} loading={projectloading} rollBackUrl="..?tab=projekty"/>

    { /* Nebo jen id porijektu a komponenta si data stáhne sama */ }
    <ProjectDetail projectId={params.projectId} rollBackUrl="..?tab=projekty"/>

    ...

    <Input
      label='Další funkcionality, komponenta ji nijak neovlivňuje ... '...
    />
  </div>
  ...
)
```

Obrázek 3.8: Použitelnost React komponenty detailu projektu

3.7.2 Detail projektu

Každý záznam o projektu si uživatel může rozkliknout a načte se mu detail zvoleného projektu. Tuto stránku jsem navrhl podle šalony abstraktu již používané pro soutěž. Zobrazené tu jsou všechny zásadní informace o projektu, zároveň jsou zde i osobní informace a kontaktní údaje, které se ale běžnému uživateli nezobrazují. Tyto citlivé údaje nejsou vůbec posílány ze serveru, aby se zamezilo odposlechnutí síťové komunikace, jejich zveřejnění a tím pádem i porušení GDPR.

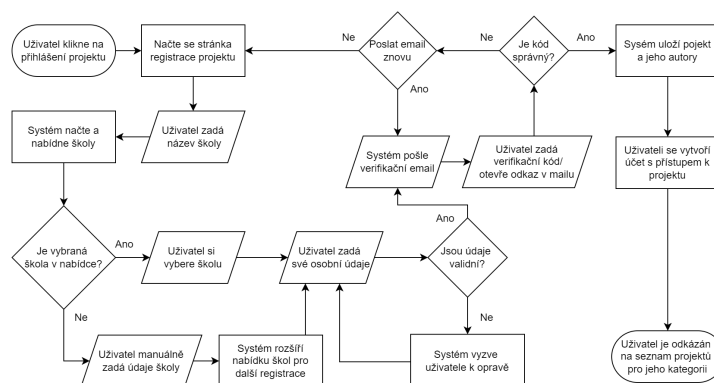
Celý detail projektu je naprogramován jako React komponenta, kterou využívám i na jiných místech v kódu. Můžu tak jednoduše vložit detail projektu na jinou stránku a nemusím se tak starat o stylování, nebo přenášení dat (viz. obrázek. 3.8).

3.7.3 Registrační formulář

Na stránkách je umístěný univerzální registrační formulář, který funguje pro všechny kola soutěže. Zájemcům o soutěž se zde nabízejí všechny otevřená kola. Každé kolo má zároveň deadline podání přihlášek a po jeho vypršení se kolo uzavře a již není soutěžícím nabízeno. Uzavření kola je potřeba kontrolovat i na backendové části. Na frontendu sice není možnost přihlásit se k uzavřenému kolu, to však nebrání poslání dotazu o přihlášení ke kolu přímo na endpoint a nezabraňovalo by to možnému "podvádění".

3.7.4 Hodnotitelská část aplikace

Každý porotce má v aplikaci svůj panel, ve kterém vidí všechny kola, do kterých byl pozván. Ty si může rozkliknout, prohlédnout si přihlášené projekty a následně mezi nimi vybrat ty, které bude během soutěže hodnotit. Tato stránka nebyla v první verzi aplikace v dokonalém stavu a po konzultaci s jednotlivým porotci jsem ji zpětně předělával, aby vyhovovala všem požadavkům. Nejvíce tak pototcům chyběla možnost hromadného přiřazování projektům a velice oblíbená zelená "fafjka", která se objeví po ohodnocení projektu.



Obrázek 3.9: Workflow registrace do soutěže

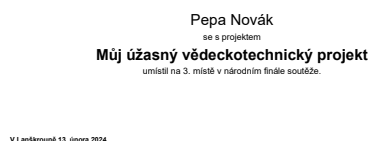
3.7.5 Administrace soutěže

Nejdůležitější a nejrozsáhlejší část systému je jeho administrace celé soutěže. Administrátor tu spravuje jednotlivé části soutěže a její kola, přístupy porotců do systému a především veškeré tiskové výstupy. Administrace obsahuje velké množství funkcí, proto níže popíšeme ty nejzásadnější. Aplikace je zároveň stále ve vývoji a je možné, že se tyto funkcionality budou od sepsání dokumentace lišit, v zájmu organizátorů soutěže.

Jako první věc, organizátor nastavuje atributy jednotlivých soutěží a kol. Tyto údaje jsou především pro účely tisku a možnosti filtrování na ostatních stránkách. Jednotlivá kola spadají pod různé soutěže (viz. kapitola Struktura soutěží) a tímto se liší jejich tiskové výstupy, mimo jiné diplomy a vizitky.

Dále se spravují konkrétní kola a to jejich datum konání, deadline zapsání a geografické umístění. Pro každé kolo je také možnost spravovat přihlášené projekty (přesunutí do jiného kola, číslování, ...) a pozvání porotců. V správě porotců zároveň vidíme projekty které hodnotí, jejich procentuelní hodnocení a komentáře k projektům. Tyto data jsou v reálném čase zobrazována během hodnocení a mohou tak sledovat průběh celé soutěže. Hodnocení projektu porotcem mohou jako admin také spravovat a to v případech, že se porotce rozhodl systém nepoužít a hodnotit na papír, jako tomu bylo v soutěži doposud.

Na základě jednotlivých hodnocení mohou nastavit výsledné umístění projektu. Toto umístění není závislé na procentuálním hodnocení jednotlivých porotců, protože na něm se porotci domlouvají společně až po ohodnocení projektů. Jednotlivá hodnocení jsou tedy pouze orientační a pomáhají rozhodováním o výsledku. Mohou si tak projekty jednoduše seřadit podle průměru procentuelních hodnocení a rozhodovat se nad výsledkem podle kvalitativní návaznosti.



Obrázek 3.10: Generování diplomu na šablonu

3.7.6 Tiskové výstupy

Po zadání výsledného hodnocení je již systém schopen vygenerovat diplomy pro celou soutěž. Pro každou možnost hodnocení je předdefinovaný text, který systém vyskočňuje podle pohlaví členů týmu. Zároveň pro každého člena vytvoří zvláštní diplom, na kterém bude právě jeho jméno na prvním místě. Toto systém dokáže jen při kliknutí na tlačítko a usnadní tím organizátorům značnou práci. Vygenerované diplomy jsou následně natisknuty na zvláštní šablonu, kterou si poskytuje soutěž sama. (viz. obrázek 3.10)

Pro tisk diplomů a ostatních tiskových výstupů jsem zvažoval několik možností a už proto, to byla časově jedna z nejnáročnějších částí aplikace. Jako první jsem se snažil o použití knihoven, ve kterých přímo upravuji PDF soubor a tím mi dovolují mít jednotnou šablonu, do které pouze doplním požadovaná data. Tato šablona by byl PDF formulář ve kterém by se vyplňovala textová pole. Jedinou nevýhodou tohoto řešení je absence responzivity jednotlivých textových polí.

Druhou zvažovanou možností bylo použití knihovny Puppeteer. Ta umožňuje spuštění virtuálního prohlížeče na backendu. Generování PDF souboru by probíhalo načtením vytvřeného souboru v HTML, pořízení snímku obrazovky a následném poslání PDF souboru jako Array Bufferu na Frontend a uložení do zařízení. Nicméně tento přístup byl nesmírně náročný na výpočetní výkon a neúměrný k účelům "jenom" generování PDF.

Nakonec jsem zvolil ironicky nejjednodušší postup a to pořízení snímku obrazovky na frontendu. To docílím nativní metodou `window.print()`, která vyvolá systémové dialogové okno tisku, ve kterém si uživatel může zvolit možnost uložit soubor do PDF.

Závěr

Vytvořil jsem plnohodnotný informační systém pro správu projektových soutěží. Systém byl otestován a úspěšně spuštěn na produkčním prostředí. Dále byl použit na okresních kolech v Lanškrouně a v Pardubicích, na krajském kole v Pardubicích a v Hradci Králové a celostátním kole v Praze. Průběžně zjišťované nedostatky byly postupně odstraňovány.

V těchto 5ti kolech jeho prostřednictvím:

- proběhlo **1023 dílčích hodnocení projektů**,
- bylo celkově **ohodnoceno 359 projektů**,
- bylo **vytištěno 485 diplomů, 590 vizitek, 14 prezenčních listin** a nepočítaně dalších tiskových výstupů

Letos provoz systému ušetřil na těchto 5ti kolech několik desítek hodin manuální práce.

Hlavní přínosy mého řešení jsou:

- Odstranění zásadních porušení GDPR,
- Centralizace pořizovaných dat, pro možnost řešení sporů a další účely,
- Automatizace tvorby všech tisků (seznamů, vizitek, diplomů, ...)
- Automatizace postupů projektů do vyšších kol soutěží,
- Snadná možnost rozšíření systému o další funkcionalitu (např. automatizace e-mailových pozvánek, automatické rozesílání personalizovaných výsledkových listin za školu, ...)

Systém přináší solidní základ pro **obecnou platformu pro pořádání projektových soutěží, která by poskytovala jejich komplexní správu**. S jeho vývojem bych rád začal co nejdříve, už proto, že jsem na jednom z kol soutěže, po odprezentování systému, dostal nabídku na zakázku pro informační systém pro další soutěž. Toto řešení by ze začátku nemuselo obsahovat veškerou funkcionalitu stávajícího systému a soustředit se hlavně na tvorbu tiskových výstupů, které byly hlavní motivací organizátorů další zmíněné soutěže.

Literatura

1. AMAVET. *O nás* [online]. [cit. 2024-03-06]. Dostupné z: http://amavet.fvtp.cz/?page_id=6.
2. AMAVET. *Festival vědy a techniky pro děti a mládež v Pardubickém kraji* [online]. [cit. 2024-03-07]. Dostupné z: https://amavet.fvtp.cz/?page_id=178.
3. AMAVET. *Výroční zprávy - AMAVET* [online]. [cit. 2024-03-19]. Dostupné z: https://amavet.fvtp.cz/?page_id=21.
4. MONIKA, Škubalová. *GDPR kontroly v České republice pro rok 2024: Zaměření Úřadu pro ochranu osobních údajů* [online]. [cit. 2024-03-20]. Dostupné z: https://www.edevizy.cz/clanek/gdpr-kontroly-v-ceske-republice-pro-rok-2024:-zamereni-uradu-pro-ochranu-osobnic_2147/.
5. VERCEL. *Next.js by Vercel* [online]. [cit. 2024-03-17]. Dostupné z: <https://nextjs.org/>.
6. PHP. *Dokumentace* [online]. [cit. 2024-03-26]. Dostupné z: <https://www.php.net/>.
7. MYSQL2. *Dokumentace* [online]. [cit. 2024-03-17]. Dostupné z: <https://sidorares.github.io/node-mysql2/docs>.
8. BCRIPT. *Dokumentace* [online]. [cit. 2024-03-19]. Dostupné z: <https://www.npmjs.com/package/bcrypt>.
9. PASSLIB. *Dokumentace* [online]. [cit. 2024-03-19]. Dostupné z: <https://passlib.readthedocs.io/en/stable/lib/passlib.hash.bcrypt.html>.
10. EXON.IO. *Next hosting* [online]. [cit. 2024-03-26]. Dostupné z: <https://exon.io/nexthosting>.
11. DOCKER. *Dokumentace* [online]. [cit. 2024-03-26]. Dostupné z: <https://docs.docker.com/>.

Seznam obrázků

1.1	Původní řešení tisku diplomů	14
1.2	Porušení GDPR původního systému	15
3.1	Možnosti přihlášení porotců	25
3.2	Metoda vytváření uživatele	26
3.3	Email provider	26
3.4	Odesílání přihlašovacích e-mailů	27
3.5	ENV proměnné pro SMTP	27
3.6	Původní use case diagram	28
3.7	Ukládání filtrů v URL adrese	29
3.8	Použitelnost React komponenty detailu projektu	30
3.9	Workflow registrace do soutěže	31
3.10	Generování diplomu na šablonu	32