



## **Bakalářská práce**

# **Inteligentní systém sledování růstu rostlin s aktivními prvky**

*Studijní program:*

B0613A140005 Informační technologie

*Studijní obor:*

Aplikovaná informatika

*Autor práce:*

**Marcel Horváth**

*Vedoucí práce:*

Ing. Jana Kolaja Ehlerová, Ph.D.

Ústav nových technologií a aplikované  
informatiky

Liberec 2024



## Zadání bakalářské práce

# Inteligentní systém sledování růstu rostlin s aktivními prvky

<i>Jméno a příjmení:</i>	<b>Marcel Horváth</b>
<i>Osobní číslo:</i>	M21000111
<i>Studijní program:</i>	B0613A140005 Informační technologie
<i>Specializace:</i>	Aplikovaná informatika
<i>Zadávající katedra:</i>	Ústav nových technologií a aplikované informatiky
<i>Akademický rok:</i>	2023/2024

### Zásady pro vypracování:

1. Proveďte rešerši komerčních i otevřených projektů pro sledování rostlin, určení jejich stavu a možnosti automatizace péče o ně.
2. Sestavte vlastní zařízení založené na Raspberry Pi se sledování zavlažování, s kamerou a připojením k síti.
3. Navrhněte vhodnou komunikaci se serverem, nainstalujte server, navrhněte a implementujte webovou aplikaci pro sledování a ovládání systému s rostlinou.
4. Zabezpečte zařízení, komunikaci, server i webovou aplikaci podle současných standardů.
5. Navrhněte možnosti rozšíření práce o využití umělé inteligence a vyšší automatizace.

*Rozsah grafických prací:* dle potřeby dokumentace  
*Rozsah pracovní zprávy:* 30 – 40 stran  
*Forma zpracování práce:* tištěná/elektronická  
*Jazyk práce:* čeština

### **Seznam odborné literatury:**

- [1] MONK, Simon. */Raspberry Pi cookbook/*. Second edition. Beijing: O'Reilly, 2016. ISBN 978-1-491-93910-9.
- [2] ANSARI, Shamshad. */Building computer vision applications using artificial neural networks: with step-by-step examples in OpenCV and TensorFlow with Python/*. For professionals by professionals. [New York]: Apress, [2020]. ISBN 978-1-4842-5886-6.
- [3] LEA, Perry. */IoT and edge computing for architects: implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security/*. Second edition. Birmingham: Packt, 2020. ISBN 978-1-83921-480-6.

*Vedoucí práce:* Ing. Jana Kolaja Ehlerová, Ph.D.  
Ústav nových technologií a aplikované  
informatiky

*Datum zadání práce:* 12. října 2023  
*Předpokládaný termín odevzdání:* 14. května 2024

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

doc. Ing. Josef Chaloupka, Ph.D.  
garant studijního programu

V Liberci dne 19. října 2023

## Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

# Inteligentní systém sledování růstu rostlin s aktivními prvky

## Abstrakt

Tato bakalářská práce se zabývá vývojem inteligentního systému pro monitorování a optimalizaci péče o rostliny. V teoretické části byla provedena rešerše komerčních, otevřených a akademických řešení, na jejichž základě bylo navrženo a sestaveno vlastní zařízení. To je postaveno na jednodeskovém počítači Raspberry Pi vybaveném řadou senzorů, čerpadlem pro zavlažování a kamerou pro pravidelné monitorování rostliny. Zařízení je integrováno do dřevěné konstrukce společně s 3D tištěnými prvky. Dále byla vyvinuta webová aplikace pro správu a ovládání tohoto systému, která umožňuje analýzu a vizualizaci dat, stejně jako přehrávání a tvorbu časosběrných videozáznamů. Následný důraz byl poté věnován nasazení a zabezpečení, aby byla zajištěna ochrana uživatelských dat a opatření proti neautorizovanému přístupu. V části práce byl také proveden pokus o rozšíření systému o prvky umělé inteligence.

**Klíčová slova:** Inteligentní pěstování rostlin, Raspberry Pi, webová aplikace, zabezpečení

# Intelligent plant growth monitoring system with active components

## Abstract

This bachelor's thesis focuses on the development of an intelligent system for monitoring and optimizing plant care. In the theoretical part, a search of commercial, open-source, and academic solutions was conducted, which formed the basis for the design and assembly of a custom device. This device is built on a Raspberry Pi single-board computer equipped with several sensors, an irrigation pump, and a camera for plant monitoring. The device is integrated into a wooden structure together with 3D-printed components. Furthermore, a web application was developed for managing and controlling this system, enabling data analysis and visualization, as well as the playback and creation of timelapse videos. Subsequent emphasis was placed on deployment and security, ensuring the protection of user data and measures against unauthorized access. An attempt was also made to extend the system with elements of artificial intelligence.

**Keywords:** Intelligent plant cultivation, Raspberry pi, web application, security

## Poděkování

Rád bych poděkoval Ing. Janě Kolaje Ehlerové, Ph.D. za vedení této bakalářské práce, všechny cenné rady a také za poskytnutí nezbytného hardwaru. Dále bych rád také poděkoval mé rodině a přítelkyni za veškerou podporu při průběhu celého studia.

# Obsah

Seznam zkratk	10
Seznam obrázků	11
Seznam tabulek	11
<b>Úvod</b>	<b>13</b>
<b>1 Současná řešení</b>	<b>14</b>
1.1 Typy zavlažování	14
1.1.1 Hydroponie	14
1.1.2 Aeroponie	14
1.1.3 Kapkové zavlažování	15
1.1.4 Aktivní zavlažování	15
1.2 Komerční řešení	15
1.2.1 Ivy	15
1.2.2 The Smart Garden 3	16
1.3 Otevřená řešení	17
1.3.1 Plantkeeper	18
1.3.2 Fyto	18
1.3.3 Raspberry Pi Powered IOT Garden	19
1.4 Akademická řešení	20
1.4.1 Systém pro automatické pěstování rostlin	20
1.4.2 Hydroponická farma ČVUT	20
1.5 Shrnutí současných řešení	21
<b>2 Použitý hardware a jeho implementace</b>	<b>22</b>
2.1 Raspberry pi	22
2.1.1 Konektivita	23
2.1.2 Instalace a konfigurace	23
2.2 Grove Base Hat	24
2.3 Aktivní prvky	24
2.3.1 Kamera	25
2.3.2 Senzor měření vlhkosti půdy	26
2.3.3 DHT senzor	26
2.3.4 Senzor světla	27
2.3.5 Displej	28
2.3.6 Zavlažování	28



2.4	LED světlo . . . . .	29
2.5	Konstrukce . . . . .	30
<b>3</b>	<b>Softwarová implementace</b>	<b>32</b>
3.1	Server . . . . .	32
3.1.1	Struktura serverového kódu . . . . .	32
3.1.2	Databáze . . . . .	33
3.1.3	Zpracování dat . . . . .	34
3.1.4	Server-sent events . . . . .	34
3.1.5	Posílání dat . . . . .	35
3.1.6	Časoběrná videa . . . . .	35
3.1.7	Posílání emailu . . . . .	37
3.1.8	Autentizace . . . . .	37
3.1.9	Testování . . . . .	38
3.2	Umělá inteligence . . . . .	38
3.2.1	Detekce zdraví rostliny . . . . .	39
3.2.2	Detekce nemoci rostliny . . . . .	39
3.3	Klient . . . . .	40
3.3.1	Struktura klientského kódu . . . . .	40
3.3.2	Styl stránky . . . . .	41
3.3.3	Axios . . . . .	41
3.3.4	Lokalizace . . . . .	41
3.3.5	Graf . . . . .	41
3.3.6	React context . . . . .	42
<b>4</b>	<b>Nasazení a zabezpečení</b>	<b>43</b>
4.1	Konfigurace serveru . . . . .	43
4.2	Docker . . . . .	43
4.3	Databáze . . . . .	43
4.4	Webový server a šifrování . . . . .	44
4.5	Nasazení aplikace . . . . .	44
<b>5</b>	<b>Cenová kalkulace</b>	<b>46</b>
<b>6</b>	<b>Testování</b>	<b>47</b>
6.1	System . . . . .	47
6.2	Webová aplikace . . . . .	47
	<b>Závěr</b>	<b>49</b>
	<b>Použitá literatura</b>	<b>50</b>
<b>A</b>	<b>Přílohy</b>	<b>53</b>

## Seznam zkratek

<b>REST</b>	Representational state transfer
<b>API</b>	Application programming interface
<b>SQL</b>	Structured query language
<b>JSON</b>	JavaScript object notation
<b>HTTP</b>	Hypertext transfer protocol
<b>URL</b>	Uniform resource locator
<b>HTML</b>	HyperText markup language
<b>CSS</b>	Cascading style sheets
<b>JS</b>	JavaScript
<b>TS</b>	TypeScript
<b>RPI</b>	Raspberry Pi
<b>LED</b>	Light emitting diode
<b>DHT</b>	Digital humidity and temperature sensor
<b>SSD</b>	Solid-State drive
<b>DNS</b>	Domain name system
<b>SSL</b>	Secure sockets layer
<b>TLS</b>	Transport layer security
<b>IP</b>	Internet protocol
<b>SSH</b>	Secure shell
<b>UFW</b>	Uncomplicated firewall
<b>CRUD</b>	Create, read, update, delete
<b>SMTP</b>	Simple mail transfer protocol
<b>UUID</b>	Universally unique identifier
<b>GPIO</b>	General purpose input/output
<b>LCD</b>	Liquid crystal display
<b>CSI</b>	Camera serial interface
<b>USB</b>	Universal serial bus
<b>HAT</b>	Hardware attached on top

## Seznam obrázků

1.1	Chytrý květináč Ivy s OLED displejem. Zdroj: Amazon, 2024 [7]. . .	16
1.2	Smart Garden 3. Zdroj: Click and Grow, 2024 [8]. . . . .	17
1.3	Celek projektu Plantkeeper. Zdroj: Emilostuff, 2020 [11]. . . . .	18
1.4	Konstrukce řešení se zavlažováním a senzory. Zdroj: Technovation, 2019 [13]. . . . .	19
1.5	Hydroponická farma Českého vysokého učení v Praze. Zdroj: Truhlařík, 2022 [15] . . . . .	21
2.1	Raspberry Pi Zero 2 WH. Zdroj: RPishop, 2024 [16]. . . . .	22
2.2	Grove Base Hat pro Raspberry Pi Zero s jeho konektory. Zdroj: Seeed Studio, 2023 [18]. . . . .	24
2.3	Raspberry Pi Camera 3. Zdroj: Botland, 2024 [19]. . . . .	25
2.4	Kapacitní senzor vlhkosti půdy. Zdroj: Botland, 2024 [19]. . . . .	26
2.5	Kapacitní senzor vlhkosti půdy - diagram. Zdroj: vlastní zpracování. .	26
2.6	DHT senzor. Zdroj: Botland, 2024 [19]. . . . .	27
2.7	DHT senzor - diagram. Zdroj: vlastní zpracování. . . . .	27
2.8	Senzor světla. Zdroj: Botland, 2024 [19]. . . . .	27
2.9	Senzor světla - diagram. Zdroj: vlastní zpracování. . . . .	27
2.10	LCD displej. Zdroj: Botland, 2024 [19]. . . . .	28
2.11	LCD displej - diagram. Zdroj: vlastní zpracování. . . . .	28
2.12	Relé. Zdroj: Botland, 2024 [19]. . . . .	29
2.13	Relé - diagram. Zdroj: vlastní zpracování. . . . .	29
2.14	USB kabel s integrovaným vypínačem. Zdroj: Aliexpress, 2024 [21]. .	29
2.15	LED růstové světlo. Zdroj: Aliexpress, 2024 [22]. . . . .	29
2.16	Diagram zapojení projektu. Zdroj: vlastní zpracování. . . . .	30
2.17	Vyrobená vlastní konstrukce. Zdroj: vlastní zpracování. . . . .	31
3.1	Schéma databáze. Zdroj: vlastní zpracování. . . . .	33
3.2	Ukázka snímku časosběrného videa. Zdroj: vlastní zpracování. . . . .	36
3.3	Vyhodnocení stavu rostliny ve webové aplikaci. Zdroj: vlastní zpracování. . . . .	40
3.4	Graf ve webové aplikaci. Zdroj: vlastní zpracování. . . . .	42
4.1	Schéma komunikace a nasazení projektu. Zdroj: vlastní zpracování. .	45

## Seznam tabulek

3.1	Pokrytí testů. Zdroj: Vlastní zpracování. . . . .	38
5.1	Cenový přehled produktů. Zdroj: Vlastní zpracování. . . . .	46

## Úvod

Automatizace a chytré technologie hrají klíčové role v dnešním světě a stávají se stále rozšířenějšími a populárnějšími. Přináší s sebou značné výhody, které umí lidem ulehčit a zefektivnit jejich každodenní úkony ve všech oblastech života. Snižují potřebu manuální a repetitivní práce od domácích prací až po komplexní průmyslové operace.

V sektoru zemědělství tyto technologie přinášejí revoluční změny. Monitorování a řízení podmínek rostlin může vést ke zvýšení kvality i kvantity úrody, využití zdrojů a tím i k zlepšení životního prostředí. Umělá inteligence může včas detekovat možné nemoci rostlin nebo nepříznivé podmínky. Kromě praktické části tato práce poskytuje přehled projektů, zaměřující se na inteligentní pěstování rostlin.

Cílem této práce je uplatnit tyto metody pěstování v domácích podmínkách a zlepšit způsob péče o rostliny. Navrhnout a implementovat systém založený na jednodeskovém počítači Raspberry Pi, který je osazen množstvím senzorů, jako senzor vlhkosti, vlhkosti půdy, teploty a osvětlení, včetně kamery pro pravidelný sběr snímků.

Součástí je vývoj webové aplikace, která bude sloužit pro monitorování a řízení celého systému. Toto webové rozhraní umožní uživatelům přístup k různým interaktivním funkcím. Uživatelé budou moci procházet jednotlivé snímky spolu s příslušnými daty ze senzorů, prohlížet grafy zobrazující dlouhodobé trendy a vytvářet časosběrná videa, která poskytují vizuální přehled o vývoji rostliny v průběhu času. Bezpečnost systému a ochrana dat je nutnost zajišťující, že osobní údaje a informace o rostlině jsou chráněny před neoprávněným přístupem a zneužitím.

# 1 Současná řešení

Existuje mnoho různých možností péče o rostliny, které se liší svými funkcemi a cenami.

Komerční produkty nabízejí různé úrovně automatizace a integrace od jednoduchých senzorů vlhkosti půdy až po komplexní systémy. Současně s rostoucí dostupností technologií mohou zdatní jedinci využít cenově dostupnější řešení, jako jsou jednodeskové počítače Raspberry Pi, Arduino nebo mikrokontrolery ESP32. To pak umožňuje sestavení vlastních systémů pro monitorování a péči o rostliny. V akademickém prostředí dochází k rozvoji inovativních prací, která často vedou k novým metodám vylepšení a technologiím v tomto směru. Tyto projekty dokážou poskytnout inspiraci pro vývoj vlastních zařízení.

Cílem je poskytnout přehled o dostupných funkcích, výhodách a nevýhodách komerčních, otevřených ale i akademických přístupů, aby se na základě této analýzy mohlo navrhnout a naprogramovat vlastní řešení, jenž bude mít pozitivní výsledky.

## 1.1 Typy zavlažování

### 1.1.1 Hydroponie

V hydroponických systémech není rostlina v přímém kontaktu s půdou. Kořeny jsou ponořené v živném roztoku, který obsahuje všechny potřebné živiny a minerály. Rostliny také potřebují ke svému uchycení substrát. Obvykle se používá keramzit, perlit nebo například kokosové vlákno. Pěstování tímto způsobem je vhodné pro pokojové rostliny, bylinky, zeleninu a ovoce.

Výhody hydroponie zahrnují delší intervaly mezi zavlažováním a umožňují rostlinám rychlejší a kvalitnější růst. Díky vyššímu obsahu vitamínů a potřebných minerálů mají rostliny menší náchylnost k nemocem a je zde také menší riziko napadení škůdci [1].

### 1.1.2 Aeroponie

V aeroponických systémech bývají rostliny volně zavěšené ve vzduchu, nejlépe v tmavých prostorech a místo běžného zavlažování jsou jen konstantně postříkovány živným roztokem. Tato metoda umožňuje pěstovat více rostlin současně. Aeroponie je sice šetrnější a rychlejší než hydroponie, ale více nákladnější na údržbu [2] [3].

### 1.1.3 Kapkové zavlažování

Kapkové zavlažování je charakteristické svou šetrností spotřeby vody. Systém zavlažuje pomocí hadic umístěných těsně u země, které kapají specifické množství vody přímo ke kořenům. Hadice má v sobě otvory, které jsou od sebe pravidelně umístěné nebo lze nainstalovat kapkovače, které regulují přesněji dávkování [4].

### 1.1.4 Aktivní zavlažování

Představuje tradiční způsob zavlažování, který vyžaduje externí zdroje energie, jako jsou čerpadla nebo rozstřikovače.

## 1.2 Komerční řešení

Na volně dostupném trhu se nacházejí chytré květináče, které nabízí jednotnou konstrukci z plastu nebo hliníku. Prodává je většina velkých elektro obchodů, jako například jsou Alza, CZC a Mall.

U některých modelů je místo běžné hlíny potřeba kupovat speciální sazenice od výrobce, které již obsahují semínka a přidané živiny. Představují větší finanční investici než normální sazenice se substrátem. I když je na výběr mnoho druhů, stále zde existují omezení v tom, co je možné pěstovat.

Většina těchto produktů jsou vybavené LED osvětlením, které fungují na specifických vlnových délkách a dodávají rostlinám dodatečné světlo, což je zvláště užitečné pro místa s nedostatkem přirozeného světla v jakémkoliv ročním období.

Ve spodní části květináče se často nachází nádrž na vodu, a tím mohou poskytnout přísun vody a usnadnit správu zavlažování.

Nabízejí také mobilní aplikace, které uživatelům posílají upozornění a umožňují nastavit interval zavlažování, ovládat osvětlení a získávat informace o rostlině a monitorovat hladinu vody v nádrži. Některé modely podporují integraci s hlasovými asistenty jako Siri, Alexa nebo Google Assistant.

Ceny chytrých květináčů začínají přibližně na 2 000 Kč, přičemž dražší modely mohou stát od 8 000 Kč výše. Ceny sazenic se obvykle pohybují mezi 250 a 350 Kč za balení [5] [6].

### 1.2.1 Ivy

Základní model chytrého květináče, který využívá běžnou hlínu. Poskytuje pouze monitorování rostliny, aniž by nabízel žádné další pokročilé funkce, jako automatické zavlažování.



Obrázek 1.1: Chytrý květináč Ivy s OLED displejem. Zdroj: Amazon, 2024 [7].

Hlavní vlastností tohoto modelu je integrace OLED displeje, který umožňuje vizuální zpětnou vazbu o stavu rostliny prostřednictvím animovaných zobrazení. Tyto animace efektivně informují uživatele o potřebách rostliny, jako je nedostatek vody, či světla [7].

### 1.2.2 The Smart Garden 3

Chytrý květináč od společnosti Click And Grow, který je populární na českém trhu, díky svému kvalitnímu designu a zpracování.





Obrázek 1.2: Smart Garden 3. Zdroj: Click and Grow, 2024 [8].

Umožňuje pěstování až tří rostlin současně. Odlišuje se od tradičních metod pěstování, jako jsou hydroponie, aeroponie nebo běžných substrátů. Využívá svoji vlastní patentovanou metodu pěstování pomocí technologie „smart soil“ neboli inteligentní substrát. Tento substrát je navržen tak, aby byl bez nečistot a bakterií, udržoval optimální pH úroveň a zajišťoval efektivní distribuci vody. Obsahuje už všechny potřebné živiny.

Kapsle jsou podobné těm, které se používají u kávovarů. Jednoduše se vloží do zařízení. Tyto kapsle obsahují již inteligentní půdu společně se semínky. Jakmile rostlina začne růst, je potřeba odstranit horní víčko, aby měla rostlina dostatek prostoru. Součástí balení jsou tři kapsle s bazalkou, další kapsle je možné zakoupit dle potřeby.

Voda se doplňuje do boční nádrže, a uživatel je upozorněn, když je potřeba nádrž znovu naplnit. Je také vybaven LED osvětlením, které se automaticky zapíná a vypíná podle časového intervalu. Veškeré nastavení a upozornění lze spravovat pomocí mobilní aplikace [9] [10].

### 1.3 Otevřená řešení

Na internetu lze najít mnoho návodů a videí ukazující jak sestavit vlastní systémy pro péči o rostliny. Tyto zdroje často obsahují 3D modely, hardwarová řešení, schémata, seznamy potřebných součástí a potřebný kód. Otevřená řešení mohou sloužit jako inspirace a poskytnout dobrý odrazový bod pro vlastní projekty.

### 1.3.1 Plantkeeper

Plantkeeper vytvořený od Emilostuff, je prezentován prostřednictvím YouTube videa, společně se všemi odkazy na potřebné materiály. Jeho systém je založen na základě Arduino desky, která má připojené kapacitní senzory vlhkosti půdy a čerpadla připojené k relé desce pro automatické zavlažování. Voda je přečerpávána z plastového boxu prostřednictvím trubic. Na tomto boxu je přidělaná 3D vytisklá konstrukce sloužící jako kryt. Z tohoto centrálního bodu vychází pak více kapacitních senzorů s trubicemi, které jsou k sobě navzájem upevněné.



Obrázek 1.3: Celek projektu Plantkeeper. Zdroj: Emilostuff, 2020 [11].

Data ze senzoru se poté posílají na platformu Blynk, která umožňuje kompletní cloudové řešení pro tvorbu webových a mobilních aplikací bez použití běžného programovacího kódu. Mobilní aplikace umožňuje uživatelům nastavit plán zavlažování, okamžitě zalít nebo získat informace o stavu rostliny na základě dat z kapacitních senzorů [11].

Tento projekt především vybočuje svým způsobem zpracování zavlažování a možností zalévat více květináčů současně, a to vše v rámci jedné atraktivní konstrukce.

### 1.3.2 Fyto

Projekt od Coders Cafe zahrnuje Raspberry Pi. Systém je naprogramován v programovacím jazyce Python. Výrobek je 3D tištěný květináč, který obsahuje integrovaný LCD displej zobrazující animované zobrazení, které odráží podobně jako u komerčních řešení „náladu“ rostliny. Zalévání rostlin je nutné provádět manuálně a neimplementuje další aplikaci pro zobrazení dat.

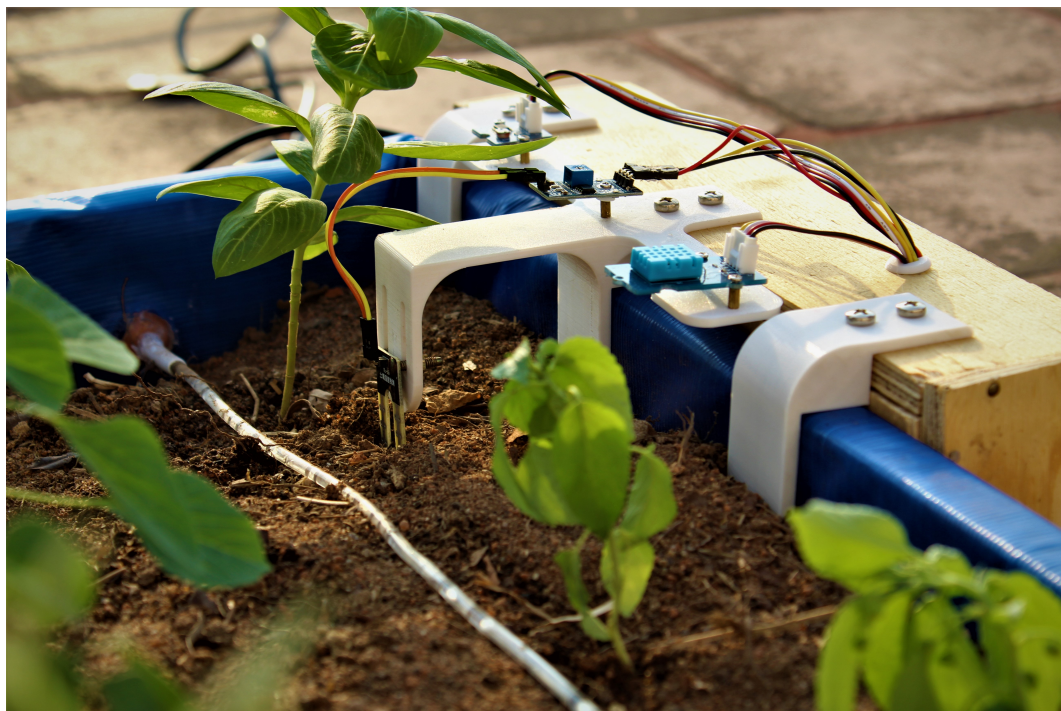
Jelikož Raspberry Pi nemá vestavěné analogové vstupy, projekt využívá A/D převodník pro převod analogových signálů ze senzorů na digitální hodnoty, které

může zpracovat. Sensory jsou kalibrovány pomocí minimálních a maximálních analogových hodnot [12].

Fyto představuje zajímavou kombinaci hardwaru a 3D tisku, která nabízí vizuálně přitažlivý způsob monitorování stavu rostlin.

### 1.3.3 Raspberry Pi Powered IOT Garden

Vyrobeno uživatelem Technovation, nabízí komplexní řešení pro inteligentní zahradu. Systém využívá Raspberry Pi osazený Grove Base Hatem, na nějž jsou připojeny důležité senzory měřící vlhkost, teplotu, světelné podmínky a vlhkost půdy. Navíc projekt zahrnuje robustní dřevěnou konstrukci pro kapénkové zavlažování, které po aktivaci relé automaticky zavlažuje celou zahradu.



Obrázek 1.4: Konstrukce řešení se zavlažováním a senzory. Zdroj: Technovation, 2019 [13].

Výsledná data jsou odesílána do databáze Firebase skrze Python. Projekt zahrnuje také vlastní android aplikaci vytvořenou s využitím MIT App Inventor, což je rozhraní, které usnadňuje tvorbu s grafickým uživatelským rozhraním. Aplikace umožňuje uživatelům sledovat a ovládat svůj systém na dálku [13].

Tento projekt představuje komplexní řešení pro automatizovanou zahradu, které je ideální pro pokročilejší uživatele. Integrace s Grove Base Hat činí tento projekt vizuálně ucelený.

Všechny tyto projekty sdílejí společný základ v podobě použití Raspberry Pi nebo Arduino desek a využívají podobných senzorů. Liší se však v přístupu ke zpracování dat a v implementaci kódů.

## 1.4 Akademická řešení

Do rešerše byla zakomponována analýza akademických řešení, aby byl poskytnut alternativní pohled na problémy a řešení v oblasti sledování rostlin.

### 1.4.1 Systém pro automatické pěstování rostlin

V digitální knihovně Dspace Technické univerzity byla nalezena pouze jedna diplomová práce od autora Tomáš Matěcha. Jedná se převážně o hardwarový systém, jehož cílem je řídit podmínky růstu ve skleníku. V první části práce analyzuje různé typy substrátů, osvětlení a zavlažovací systémy. Druhá část je věnována popisu návrhu a realizace.

Kombinuje hardwarové a softwarové komponenty pro komplexní řízení skleníkových podmínek. Tento modulární systém komunikuje přes sběrnici RS485 a skládá se ze dvou hlavních modulů. Řídícího, který je vybaven LCD a ovládáním pro nastavení a měřícího, který je zaměřen na snímání teploty a vlhkosti. Tento měřící modul aktivuje akční prvky na základě příkazů od řídicího modulu, což umožňuje správu klimatu skleníku.

Použitý zavlažovací systém v Matěchově práci je založen na aeroponické závlaze, ale v jeho realizaci se ukázala jako méně efektivní pro mladé rostliny, kvůli nedostatečnému zavlažování kořenového systému. Tento náález je užitečný pro tuto práci, jelikož zdůrazňuje potřebu detailního výběru a adaptace zavlažovacího systému [14].

Na rozdíl od Matěchovy práce, tato bakalářská práce zahrnuje integraci komplexního softwarového řešení, včetně vývoje webové aplikace, což poskytuje rozšířené možnosti a také příjemnější sledování podmínek rostlin.

### 1.4.2 Hydroponická farma ČVUT

Na Fakultě elektrotechnické Českého vysokého učení technického v Praze byla vytvořena hydroponická farma, která využívá zavlažování typu „flood and drain“. Tato technika periodicky naplní a vypouští dvě vany s vodou dle potřeb rostlin, což zajišťuje vlhkost pro kořeny a kontrolu nad cyklem zalévání. Farma dokáže pěstovat až pět rostlin najednou. Celý tento systém řídí deska Arduino Mega 2560.

Bakalářská práce Václava Truhlaříka na tento projekt navazuje a rozšiřuje ho o modul ESP32 a ESP32 kameru. Kamera má dvě základní funkce: pravidelné ukládání snímku na SD kartu pro tvorbu časosběrných videí, ale ty se musí tvořit zvlášť na vlastním počítači pomocí projektu ffmpeg nebo ke streamování živého obrazu přes webový server. Tento server je vyvinut programovacím jazykem C++.

V práci jsou rovněž popsány hydroponní systémy a proces vytváření osvětlení s použitím RGB modulu a vlastnoručně tištěné 3D konstrukce [15].



Obrázek 1.5: Hydroponická farma Českého vysokého učení v Praze. Zdroj: Truhlařík, 2022 [15]

Jedná se o komplexní projekt, jehož přínosem je využití kamery. Pro zlepšení by bylo vhodné využít více senzorů pro lepší monitorování a vytvoření pokročilejší webové aplikace.

## 1.5 Shrnutí současných řešení

V této rešerši byly zkoumány a srovnávány různé metody péče o rostliny, přičemž byl kladen důraz na porovnání komerčních, otevřených a akademických řešení. Komerční produkty nabízejí snadnou instalaci a spolehlivost, avšak jsou často spojeny s vyššími náklady. Nicméně otevřená řešení mají větší prostor pro přizpůsobení a experimentování, což je výhodné pro technicky zdatné uživatele s dostatečným časem. Akademická řešení přinášejí nové myšlenky, hlubší pochopení dané problematiky a využití širokého spektra zdrojů.

Chytré květináče, i přes vysokou počáteční investici, poskytují výhody zjednodušené až zautomatizované péče a zlepšení kvality rostlin, oproti tradičnímu pěstování.

Každý z přístupů má své specifické výhody a nevýhody, a výběr nejvhodnějšího řešení tak závisí na individuálních potřebách. Z této rešerše vyplynulo, že kombinace prvků z každého přístupu by mohla vést k vytvoření optimalizovaného a efektivního systému pro péči o rostliny.

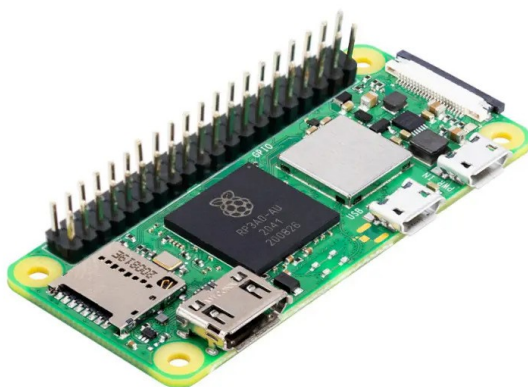
## 2 Použitý hardware a jeho implementace

V této kapitole se popíše hardwarové komponenty, jejich zapojení, instalace a případné řešení problémů, které při vývoji nastaly.

### 2.1 Raspberry pi

Raspberry pi je malý jednodeskový počítač založený na ARM architektuře, která má využití například u mobilních telefonů, kvůli nízké spotřebě a vysokému výkonu. Jeho první verze byla vytvořena v roce 2012 společností Raspberry Pi foundation. Díky své cenové dostupnosti od 100 do 2500 korun se stal velice populární u programátorů a IT nadšenců při tvorbě mnoha projektů.

Existuje několik verzí. Hlavní modely Raspberry Pi se číslují 1 až 5, přičemž Raspberry Pi 5 je nejnovější a nejvýkonnější verze. Tyto modely se dále dělí podle množství RAM. Kromě těchto základních verzí nabízí se ještě menší a méně výkonné varianty, jako je Raspberry Pi Zero 1 a 2 nebo nejlevnější verze Raspberry Pi Pico, která je určena jako mikrokontrolér.



Obrázek 2.1: Raspberry Pi Zero 2 WH. Zdroj: RPishop, 2024 [16].

### 2.1.1 Konektivita

Pro napájení je doporučeno používat stabilní 5V zdroj. Podle množství připojených periférií a modelu je typický proud od 1 A až 3 A.

Raspberry Pi je vybaveno různými porty jako USB, HDMI, micro USB, slotem pro SD kartu, konektorem pro kameru a v případě nejnovějších modelů také slotem pro PCI express.

RPI lze také vybavit GPIO headery, které umožňují připojení velkého množství senzorů a modulů. Využívá digitální signál, pro čtení analogových signálů je potřeba analog-digitální převodník. Při špatném zacházení s těmito piny, může dojít k poškození celého zařízení.

Jako úložiště Raspberry Pi se typicky používá SD karta, která má dedikovaný slot. Na tuto kartu se nahrává také operační systém. Je vhodné pořídit si kvalitní SD kartu kvůli rychlosti čtení a zápisu, což přispívá k celkové svižnosti systému. Na RPI lze použít libovolný podporovaný systém, nicméně pro nejlepší kompatibilitu a nejsnadnější instalaci je doporučeno si stáhnout na svém hlavním zařízení z webových stránek oficiální program Raspberry Pi Imager, který nainstaluje libovolnou distribuci Linuxu a umožňuje také konfiguraci. RPI má svojí vlastní Linux distribuci jménem Raspbian nebo Raspberry Pi OS. Tento operační systém je navržen přímo na tento typ hardwaru. Poté se SD karta vloží do RPI, které se automaticky naboootuje [17].

K ovládání RPI je možné pracovat s hardware perifériemi, jako je klávesnice, myš a monitor. V důsledku použití RPI Zero 2 je efektivnější a rychlejší připojit se k zařízení vzdáleně pomocí protokolu SSH a ovládat ho přes příkazovou řádku. Z tohoto důvodu byla nainstalovaná pouze verze operačního systému bez grafického rozhraní, to přispívá také ke snížení systémových zdrojů. SSH protokol se nastaví a povolí při instalaci v Raspberry Pi Imageru.

### 2.1.2 Instalace a konfigurace

Pro účely této práce se využívá model Raspberry Pi Zero 2 W, který byl k dispozici a který poskytuje dostatečný základní výkon pro odesílání dat na server, zároveň obsahuje integrované Wi-Fi. Pokud by byl k dispozici výkonnější model, mohl by sloužit pro celkové nasazení systému.

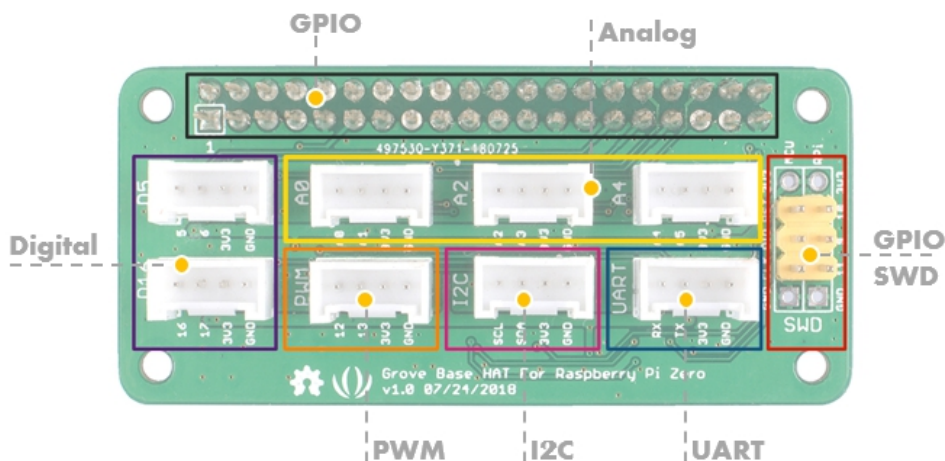
Na RPI byla nainstalována oficiální lite verze operačního systému Raspberry Pi OS prostřednictvím jejich imageru, který nainstaloval systém na SD kartu. Tato verze neobsahuje grafické uživatelské rozhraní, proto je potřeba provádět všechnu práci v terminálu. Jako první bylo potřeba zajistit připojení zařízení k internetu. To se ve vývojové části zajistilo tak, že se spustil mobilní hotspot na počítači, na kterém se vytvářela serverová část, a proto bylo následně možné se připojit prostřednictvím SSH. Hlavním důvodem byla častá změna umístění.

Po základním nastavení byl systém aktualizován a doinstalován potřebný software - Python, libcamera a další. Protože RPI nemá vlastní hodiny reálného času pro udržení času po vypnutí, používá se Network Time Protocol pro synchronizaci času přes internet. Kvůli nedostupnosti internetu při startu systému byl navíc

vytvořen cron démon, umožňující manuální nastavení času.

## 2.2 Grove Base Hat

Jedná se o rozšiřující desku od firmy Seeed Studio, která se nasazuje přímo na GPIO RPI. Disponuje digitálními, I2C, UART, PWM porty a díky integrovanému mikrokontroleru STM32 také analogovými porty [18].



Obrázek 2.2: Grove Base Hat pro Raspberry Pi Zero s jeho konektory. Zdroj: Seeed Studio, 2023 [18].

K těmto konektorům se dá připojovat jednoduše přes čtyř pinové kabely. Grove nabízí celkový ekosystém. Umožňuje připojení vysoké škály hardwarových modulů jako jsou senzory, tlačítka, displeje a mnoho dalšího.

Velká výhoda je, že se jedná o jednoduché řešení, a proto není potřeba hluboké znalosti hardwaru, pájení nebo nepájivého pole. Disponuje taky rozšířením pro GPIO piny. Uživatelé se tak nemusí obávat, že přijdou o tuto konektivitu.

Společnost poskytuje rozsáhlou podporu pro vývojáře a komunitu, včetně aktivních sociálních sítí, fór a podrobně dokumentované Wiki. Na těchto platformách je možné nalézt návody, dokumentaci a příklady kódu. Na Githubu jsou k dispozici příkladové kódy napsané v Pythonu nebo C++.

## 2.3 Aktivní prvky

Ke sledování růstu rostlin je zapotřebí několika senzorů a hardware komponent. Grove disponuje širokou škálou těchto prvků, které jsou pro tento účel vhodné.

Pro izolaci závislostí v adresáři s projektem bylo vytvořeno Python virtuální prostředí, kam se nainstalovali všechny knihovny. Mezi hlavní využívané knihovny



patří picamera, RPI.GPIO, requests a také oficiální knihovny od Seeed studia, které se museli nainstalovat manuálně podle návodu.

Zdrojový kód je rozdělen do několika Python souborů podle dané funkcionality. Jeden z těchto kódů zajišťuje komunikaci se senzory. Obsahuje třídy pro každý potřebný aktivní prvek s metodami pro sběr a formátování dat. Druhý řeší psaní zobrazení dat ze senzorů na displej. Třetí se stará o práci s kamerou. Hlavní soubor spojuje tyto třídy dohromady a to znamená že, pravidelně čte data ze senzorů, pořizuje fotografie, zavlažuje, zobrazuje data na displej a odesílá na server.

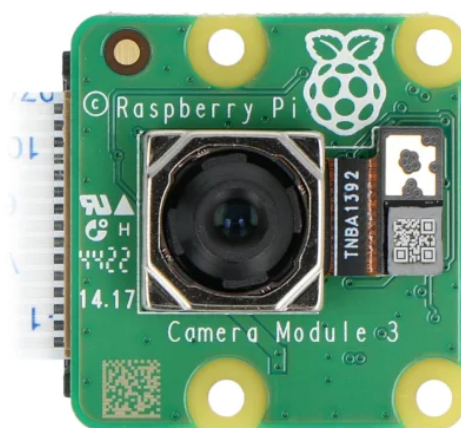
RPI všechna data ze senzorů odesílá na server prostřednictvím REST API. K odesílání obrázku se využívají MIME typy, které umožňují identifikovat formát souborů a na základě jeho přípony ho správně formátovat. Následně je přidán do těla požadavku jako multipart/form-data. Po úspěšném odeslání se uložený obrázek na RPI smaže. Také pravidelně posílá GET požadavky na server, aby zjistil, jak často odesílat data nebo zda má spustit zavlažování. Tohle vše se dá nastavit ve webové aplikaci na stránce nastavení. K bezpečné komunikaci se serverem je přikládán do každého požadavku REST API token.

### 2.3.1 Kamera

K RPI lze připojit kameru pomocí CSI konektoru. Při zapojování je důležité mít orientovaný plochý kabel na správnou stranou. Pro RPI Zero 2 je nutné si pořídit užší verzi plochého kabelu, kvůli menšímu CSI portu na tomto zařízení.

Na začátku tvorby projektu byl proveden pokus o použití kamery Arducam OV5647, přes mnoho pokusů a různých konfigurací nebylo možné tuto kameru zprovoznit. Problém mohl být z důvodu nekompatibility ovladačů nebo fyzickému poškození.

Kvůli tomuto problému byla zakoupena originální kamera - Raspberry Pi Camera 3, která fungovala zcela bez problémů. Jedná se o 12mpx senzor, který má vysoké rozlišení a automatické zaostřování narodil od kamery Arducam.



Obrázek 2.3: Raspberry Pi Camera 3. Zdroj: Botland, 2024 [19].

Pro správné fungování je nutné nainstalovat balíček libcamera-apps. U starších verzích operačního systému balíček raspistill a raspivid a také povolit rozhraní kamery v nastavení raspi-config. Systém by měl být aktualizován na nejnovější verzi příkazy `sudo apt update` a `sudo apt upgrade`.

Pythonu třída pro ovládání kamery nejprve inicializuje nastavení kamery, kde je možné určit adresář pro ukládání fotografií a případně nastavit manuální zaostření. Dále bylo nutné upravit výchozí nastavení logování tak, aby se vypisovaly pouze případné chyby. Funkce pro pořízení fotografie pak uloží snímek na disk a po úspěšném odeslání na server jej vždy smaže.

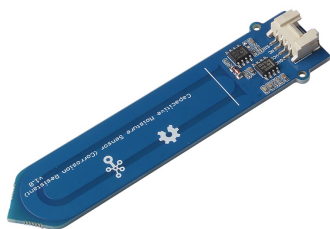
### 2.3.2 Senzor měření vlhkosti půdy

Tento senzor umožňuje měření procentuální vlhkosti půdy díky analogovému signálu. Sensory vlhkosti půdy můžeme obecně rozlišit na dva typy: rezistivní a kapacitní.

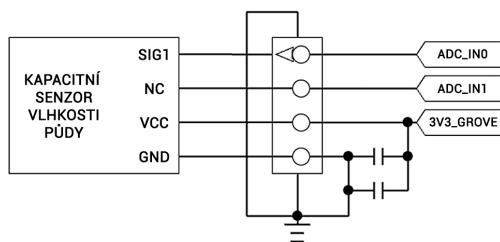
Rezistivní senzory se skládají ze dvou vodivých částí a měří vlhkost půdy na základě odporu mezi nimi. Jsou však náchylné k erozi a mají kratší životnost při kontinuálním používání.

Na druhou stranu, kapacitní senzory jsou odolnější vůči korozi a nevyžadují častou údržbu. Fungují na principu změny kapacity mezi dvěma vodivými povrchy, která se mění s vlhkostí půdy [20].

Seed studio nabízí oba tyto senzory. Pro lepší měření a menší údržbu byl zvolen kapacitní senzor.



Obrázek 2.4: Kapacitní senzor vlhkosti půdy. Zdroj: Botland, 2024 [19].



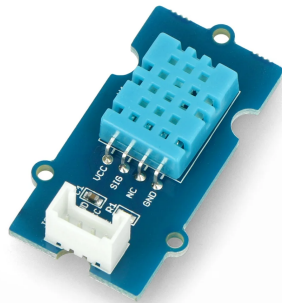
Obrázek 2.5: Kapacitní senzor vlhkosti půdy - diagram. Zdroj: vlastní zpracování.

Po připojení senzoru k analogovému portu Grove Base Hat A0 byla provedena kalibrace, aby se získal procentuální výstup založený na hodnotách, kdy senzor byl suchý a kdy byl ponořený ve vodě. Třída senzoru obsahuje funkce jak pro surový výpis signálu, tak i procentuální, využívajíc při tom kalibrační hodnoty 590 a 330. Analogový signál je čten pomocí knihovny `grove.adc`.

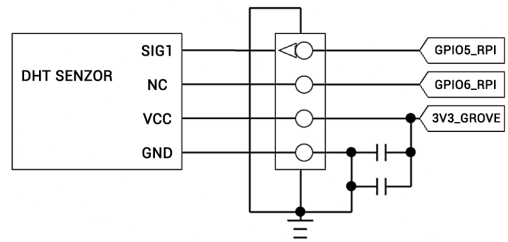
### 2.3.3 DHT senzor

Digitální senzor měří teplotu a vlhkost vzduchu a je předem kalibrován. Data jsou dostupná ve stupních Celsia a procentuální vlhkosti. Pro práci s tímto senzorem

byla použita Python oficiální knihovna od Seeed Studio.



Obrázek 2.6: DHT senzor. Zdroj: Botland, 2024 [19].

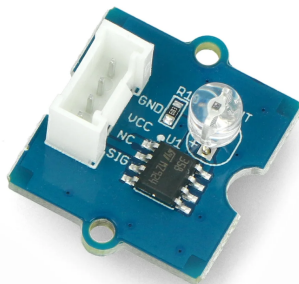


Obrázek 2.7: DHT senzor - diagram. Zdroj: vlastní zpracování.

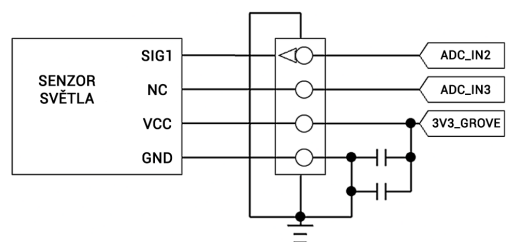
Získání teploty a vlhkosti z DHT senzoru závisí na knihovně `seed_dht`, která se instaluje samostatně. Tato knihovna obsahuje předem kalibrované hodnoty, což umožňuje čtení dat ze senzoru bez nutnosti dalšího nastavování. Senzor je připojen k digitálnímu portu D5.

### 2.3.4 Senzor světla

Tento senzor se skládá z fotoresistoru a je tedy analogový. S rostoucím světlem se zvyšuje jeho analogová hodnota. Senzor byl také kalibrován ve tmě a na přímém světle. Poskytuje přesné měření a je velmi citlivý.



Obrázek 2.8: Senzor světla. Zdroj: Botland, 2024 [19].



Obrázek 2.9: Senzor světla - diagram. Zdroj: vlastní zpracování.

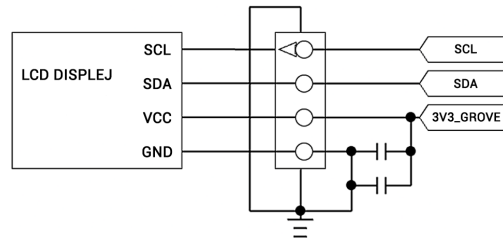
Podobně jako senzor vlhkosti půdy analogový signál je čtený skrze knihovnu `grove.adc` z analogového vstupu A2. Pro kalibraci byly naměřeny hodnoty ve tmě, kde byla hodnota 0, a při přímém osvětlení, kde byla hodnota 702.

### 2.3.5 Displej

Kromě zobrazení dat ze senzorů ve webové aplikaci se data zobrazují také na displeji. Používá se 2x16 LCD displej s podsvícením, který se připojuje pomocí jednoho kabelu přes I2C rozhraní do Grove Base Hatu.



Obrázek 2.10: LCD displej. Zdroj: Botland, 2024 [19].



Obrázek 2.11: LCD displej - diagram. Zdroj: vlastní zpracování.

Pro snadnou manipulaci s tímto displejem byla vytvořena Python třída. Tato třída využívá knihoven GPIO a smbus pro komunikaci s I2C sběrnici, což umožňuje odesílání bajtů obsahujících příkazy pro zobrazení textu nebo čištění displeje.

### 2.3.6 Zavlažování

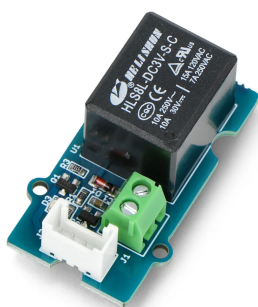
Automatické zavlažování funguje na základě aktivního systému, který čerpá vodu z nádoby pomocí hadice do květináče rostliny.

V tomto zavlažovacím systému je použito relé, které je připojeno k digitálnímu portu D16 na Grove modulu. Když je relé aktivováno, dojde k uzavření obvodu, který umožňuje provoz čerpadla. Relé je schopné spínat napětí až do 250V při proudu do 10A. Aktivace je indikována červenou LED diodou.

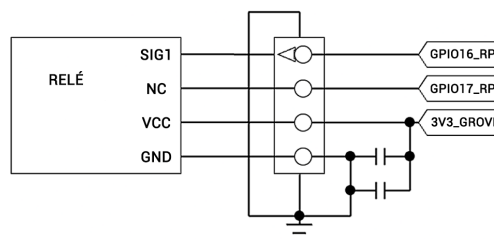
Napájení čerpadla a RPI je zajištěno z jediného zdroje. Kvůli vysokým energetickým požadavkům čerpadla, které překračují kapacitu RPI, bylo nutné přidat odbočku v napájecím kabelu. Tato odbočka umožňuje přímé napájení čerpadla z hlavního zdroje, zatímco sekundární větve vede energii do RPI.

Čerpadlo vyžaduje stejnosměrné napětí ve výši 3V až 5V. Ačkoliv je napětí poskytované jedním GPIO pinem dostatečné (přibližně 3,3V), každý GPIO pin může bezpečně dodávat pouze proud okolo 16 mA, což je nedostatečné pro provoz čerpadla a mohlo by potenciálně způsobit poškození RPI.

Toto čerpadlo poskytuje dostatečnou sílu pro průtok vody k zavlažování rostliny.



Obrázek 2.12: Relé. Zdroj: Botland, 2024 [19].



Obrázek 2.13: Relé - diagram. Zdroj: vlastní zpracování.

Třída zodpovědná za řízení relé ho dokáže zapnout, vypnout, spustit na nastavenou dobu, kontrolovat jeho stav a při ukončení skriptu nebo v případě chyby provede odpojení GPIO pinů, čímž zabraňuje potenciálnímu poškození.

## 2.4 LED světlo

Pokud se bude počítat s tím, že rostlina nebude na místě s dostatečným denním světlem, bylo zakoupeno 5V LED růstové světlo. To funguje na přesné světelné frekvenci, napomáhající ke zdravému růstu.



Obrázek 2.14: USB kabel s integrovaným vypínačem. Zdroj: Aliexpress, 2024 [21].

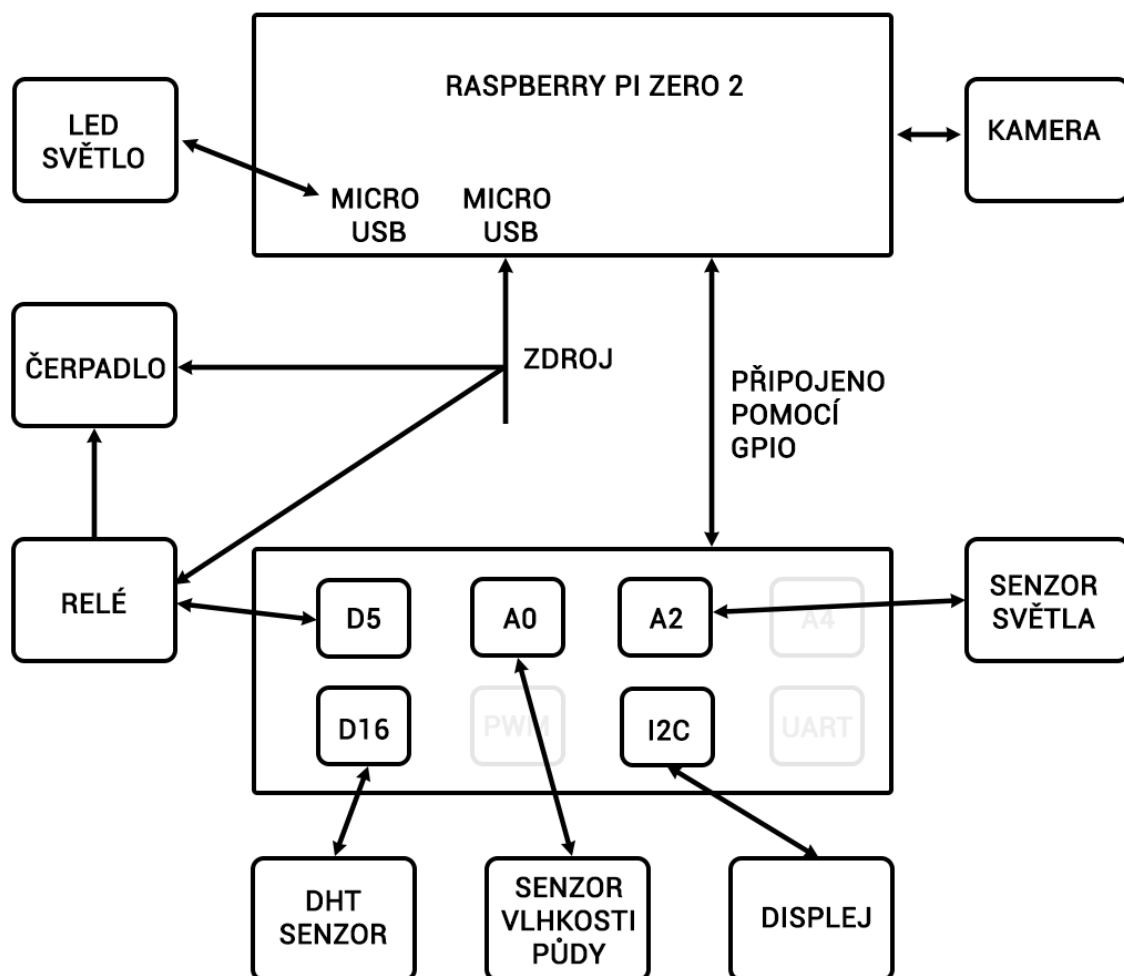


Obrázek 2.15: LED růstové světlo. Zdroj: Aliexpress, 2024 [22].

RPI Zero 2 neumožňuje softwarové zapínání či vypínání napájení u micro USB portů, aby umožnilo automatizované ovládání osvětlení. K ovládání LED světla

byl proto použit USB kabel s integrovaným vypínačem, který alespoň umožňuje manuální kontrolu nad světlem bez nutnosti odpojování z napájecího zdroje.

## 2.5 Konstrukce



Obrázek 2.16: Diagram zapojení projektu. Zdroj: vlastní zpracování.

Schéma poskytuje ucelený přehled o rozložení a připojení všech hardwarových komponentů tohoto projektu. Zobrazuje jak jsou všechny aktivní prvky zapojené a dále je naznačeno řešení pro napájení RPI a zavlažovacího systému. Slouží jako orientační mapa.



Obrázek 2.17: Vyrobená vlastní konstrukce. Zdroj: vlastní zpracování.

K uspořádání hardwaru do praktického strukturovaného celku, byla navržena vlastní konstrukce. Základem projektu je dřevěná bedýnka, ke které jsou vertikálně připevněné dvě dřevěné latě, jež udržují kameru a zdroj světla ve správné poloze.

S ohledem na ochranu hardwaru bylo využito 3D tisk. Modely byly staženy nejdříve z internetu a upraveny programem Blender, aby vyhovovali specifikacím projektu. Byl vytisknut obal pro RPI, do kterého se vejde také Grove Base Hat. Obsahuje prostor pro vývod kabelů, včetně připojení pro kameru. Kryt kamery funguje jako držák s možností menšího přizpůsobení úhlu snímání. Pro DHT a světelný senzor byly rovněž vytisknuty jednoduché obaly. Všechny tyto 3D výtisky jsou připevněné ke dřevěné konstrukci.

Květináč s rostlinou společně s nádobou na vodu a čerpadlem jsou umístěny v bedýnce. Toto řešení tvoří jednotný systém, který zároveň minimalizuje riziko poškození důležitých součástí.

## 3 Softwarová implementace

Při vývoji tohoto systému bylo využito mnoha moderních technologií. K tvorbě webové aplikace byl hlavně použit programovací jazyk Typescript, zatímco skripty běžící na RPI byly programovány v Pythonu, který sloužil pro komunikaci s hardwarem a posílání dat na server. Kromě toho byl Python také použit i pro tvorbu modelu umělé inteligence.

### 3.1 Server

Serverová část aplikace je postavena na prostředí Node.js společně s frameworkem Express.js. Tato kombinace umožňuje tvorbu rychlého asynchronního serveru založeného na komunikaci pomocí REST API. Celý kód je napsán v TypeScriptu, což zlepšuje typování a čitelnost kódu.

#### 3.1.1 Struktura serverového kódu

Hlavním souborem projektu je `app.ts`, který slouží jako vstupní bod serveru. V tomto souboru jsou importovány všechny potřebné routy (API cesty) a další konfigurace.

Každé route je možné přiřadit kontrolér, které zpracovává příchozí požadavky a generuje odpovědi. Navíc, routy mohou využívat middleware k zpracování požadavků ještě před jejich dosažením kontrolérů, což umožňuje, například, autentizaci uživatelů.

Projektová struktura zahrnuje několik klíčových adresářů:

- **Routes:** Definuje cesty API a přiřazuje k nim odpovídající kontroléry a middleware.
- **Kontroléry:** Obsahuje soubory, které obsahují logiku spojenou s jednotlivými API cestami a zpracovávají dané požadavky.
- **Middleware:** Nachází se zde middleware funkce, pro zpracování požadavků a odpovědí.
- **Utils:** Tento adresář obsahuje pomocné metody pro formátování, práci s obrázky a videi, a také cron úlohy.
- **Prisma:** Adresář pro správu databáze s využitím Prisma. Obsahuje schéma databáze a migrační soubory.



- **Static:** Zde je uložen všechny mediální obsah, jako jsou obrázky rostlin a časosběrná videa.

Konfigurace projektu a seznam nainstalovaných Node.js balíčků lze najít v souboru package.json.

### 3.1.2 Databáze

Správa databáze je postavena na objektivě relačním mapování Prisma. Ta výrazně zjednodušuje práci s databází. Umožňuje tvorbu modelů přímo v serverovém kódu s typovou bezpečností. Pomocí Prisma je možné definovat, migrovat a spravovat databázové schéma, což zahrnuje jak tvorbu nových tabulek, tak i aktualizaci existujících struktur bez přímého zásahu do jazyka databáze. Podporuje vysokou škálu databází, a to z ní činí univerzální nástroj pro práci s daty [23].

V kódu se pak na Prismu jednoduše odkazuje pro vykonávání dotazů na databázi, což dělá interakci s databází rychlou a bezpečnou. Bez nutnosti znalosti syntaxe dané databáze.

Ve vývojové fázi byla z důvodu jednoduchosti a nenáročnosti na hardware použita databáze SQLite. Ukládá záznamy do jediného souboru. V případě omezených zdrojů je toho vhodná volba pro málo výkony hardware. Z důvodu nasazení serveru zvláště na virtuální stroj, byla jako databáze v produkci zvolena MySQL, která je robustnější.

SensorData	
id	nvarchar(36) PK
soilMoisture	decimal(5,3)
temperature	decimal(5,3)
humidity	decimal(5,3)
light	decimal(5,3)
createdAt	datetime(3)

TimelapseData	
id	nvarchar(36) PK
thumbnail	nvarchar(100)
createdAt	datetime(3)

plantSettings	
id	nvarchar(36) PK
captureInterval	int
wateringDuration	int
waterPlant	tinyint(1)

emailSettings	
id	int PK
cronTime	nvarchar(5)
subject	nvarchar(100)
recipient	nvarchar(50)

user	
id	int PK
username	nvarchar(100)
password	nvarchar(100)

ApiKey	
id	int PK
key	nvarchar(100)
usageCount	int
lastUsed	datetime(3)
createdAt	datetime(3)



Obrázek 3.1: Schéma databáze. Zdroj: vlastní zpracování.

Projekt potřebuje několik modelů. Nejdůležitějším modelem je tabulka pro uchování všech dat ze senzorů, datum a identifikátor, který zároveň odkazuje na název obrázku. Poté je zde tabulka pro tvorbu časosběrných videí a nakonec pomocné tabulky jako nastavení emailu, konfigurace RPI a tabulky pro zabezpečení.

### 3.1.3 Zpracování dat

Po odeslání sensorových dat a fotografie z RPI na server se požadavek zpracovává v několika krocích. Při příjmu požadavku se prvně kontroluje, jestli je přítomen správný REST API token v hlavičce x-api-key a porovnává se s uloženými tokeny v databázi. To zaručuje, že požadavek pochází z autorizovaného zdroje. Jako další se ověřuje zda-li přijatá data ze sensorů jsou platná a kompletní. Ačkoli tato validace probíhá již na straně RPI, serverová kontrola poskytuje další úroveň zabezpečení proti potenciálně chybným nebo neúplným datům.

Každému záznamu a obrázku se přiřazuje UUID, který zaručuje, že každý záznam i obrázek je jedinečný.

Přijatý obrázek z RPI je zpracován balíčkem Jimp, který vytváří miniaturu snížením kvality a rozlišení původního obrázku. Tato miniatura poskytuje rychlejší načítání a zobrazení náhledu uživatelům ve webové aplikaci. Originální obrázek, tak jeho miniatura jsou uloženy ve složkách static a dostupné pro zobrazení pomocí url. V případě jakékoliv chyby během zpracování nebo ukládání dat se příslušný záznam a všechny soubory (obrázky a miniatury) smažou, aby se zajistila konzistence dat. Server dále monitoruje dostupné úložiště a v případě jeho nedostatku automaticky čistí menší množství starších uložených dat a obrázků.

### 3.1.4 Server-sent events

Aby klientská část aplikace měla vždy přístup k nejaktuálnějším datům pořízených z RPI, byla implementována technologie Server-Sent events.

Funguje to tak, že když se klient připojí k serveru, naváže otevřené spojení prostřednictvím specifických HTTP hlaviček. Kvůli tomu zůstává komunikační kanál mezi serverem a klientem aktivní, a server tak může posílat zprávy a aktualizace všem klientům.

- **Content-Type: text/event-stream:** Informuje klienta, že server bude posílat data ve formátu event-stream.
- **Cache-Control: no-cache:** Zabraňuje ukládání odpovědi do mezipaměti.
- **Connection: keep-alive:** Udržuje spojení otevřené bez nutnosti opakovaného navazování spojení.

Pravidelně každých pár sekund server odesílá všem připojeným klientům signál, který udržuje spojení aktivní a zabraňuje jeho časovému vypršení. Spojení zůstává aktivní, dokud klient aplikaci nezavře nebo dokud nedojde k přerušení spojení z jiných důvodů jako například ze ztráty síťového připojení. Jestliže toto nastane, klient je odstraněn ze seznamu aktivních uživatelů a spojení se ukončí.

Kromě signálu pro vypršení, server také odesílá zprávy o nově příchozích datech z RPI. Jakmile server obdrží nová data, rozešle zprávu všem klientům, kteří následně po přijetí vědí že si data mohou aktualizovat.

Tento způsob implementace je jednodušší než použití WebSockets a pro tento projekt dostačující. Server-Sent Events umožňuje jednosměrnou komunikaci ze

serveru pomocí HTTP a zajišťuje, že klientská část má vždy k dispozici aktuální data rostliny.

### 3.1.5 Posílání dat

Server poskytuje řadu REST API cest, pro konfiguraci a zobrazení dat. Uživatelé mohou získat specifická data několika způsoby: vyhledáváním podle identifikátoru k získání konkrétního záznamu, vyhledáváním podle data nebo získání posledního porízeného záznamu.

Pro orientaci v záznamech je zde API cesta, která slouží k procházení miniatur a využívá se jako navigace na klientské části. Zásluhou této cesty lze najít díky identifikátoru podrobnější informace o rostlině.

Tyto API cesty obsahují odkazy na obrázek a miniaturu rostliny. Ty jsou uloženy ve static složce, jež je konfigurována tak, aby automaticky servírovala svůj obsah. Pro zobrazení obrázku v uživatelském rozhraní stačí vložit příslušný odkaz do HTML tagu `img`.

K zobrazení na grafu jsou data získaná ze senzorů předzpracována, aby byla zajištěna optimalizace na klientské části. Nejdříve se všechny data rozdělí do svých vlastních polí a v případě vysokého množství, se automaticky shlukují pro menší zátěž, ale i zároveň k udržení přehlednosti grafu. V případě, že data překračují rozsah od sebe více jak tři dny, zobrazí se pouze každý desátý záznam.

Kromě toho server má API cesty pro tvorbu a správu časosběrných videí. Dále jsou k dispozici cesty pro konfiguraci Raspberry Pi, správu REST API klíčů a autentizaci.

### 3.1.6 Časosběrná videa

Vytváření videí je založeno na nástroji FFmpeg. Jedná se o program, který je multiplatformní a umožňuje široké množství funkcí pro práci s různými médii v prostředí terminálu.

FFmpeg musí být nainstalován na zařízení, kde server běží. Ke komunikaci s tímto programem je využíván Node.js balíček `Fluent-ffmpeg`, který vyhledá spustitelný soubor FFmpeg na základě systémové proměnné prostředí `PATH`. Nejprve je potřeba vytvořit jeho konstruktor, který vykonává příkazy FFmpeg skrze `Typescript`. To umožňuje abstrakci a jednoduchou integraci v kódu.

Proces tvorby videí je velice komplexní a zahrnuje množství parametrů, které lze nastavit. Kontrolér očekává následující: počet snímku za sekundu, rozlišení videa a pravdivostní hodnoty pro překryv grafem nebo data s časem. Všechny tyto vstupní parametry se validují.



Obrázek 3.2: Ukázka snímku časoběrného videa. Zdroj: vlastní zpracování.

Základ videa tvoří snímky, které byly pořízené na RPI. Pro základní tvorbu videa se nejdříve naleznou všechny záznamy podle rozsahu data a času a také fotky podle připadajících se identifikátorů. Cesty ke snímkům se zapíší do dočasného souboru, aby se minimalizovalo množství dat udržovaných v paměti. Tento soubor pak slouží jako vstup pro FFmpeg. Použijí se požadavky uživatele a podle nich se nastaví dané snímky za sekundu, výstupní rozlišení, včetně dalších fixních parametrů pro vytvoření videa, jako je kodek a formát pixelů výstupního videa. Pro toto nastavení byli použité běžné kombinace těchto parametrů při tvorbě videí. Byl zvolen kodek libx264, jenž uchovává dobrou kvalitu videa s nízkou velikostí souboru a jako formát pixelu byl vybrán yuv420P.

Vytvořené video se uloží na disk a informace o něm jsou zaznamenány v databázi. Po dokončení se smažou všechny dočasné složky a soubory.

Překrytí videa či daty s časy se vytváří podobným způsobem. Překrytí grafem je výkonově a časově náročnější na tvorbu, jelikož se každý snímek grafu musí vygenerovat balíčkem Chartjs-node-canvas. Tyto snímky se ukládají do dočasné složky. Poté proces tvorby videa je tvořen podobným způsobem jako u toho základního. Když jsou obě videa vygenerována kombinují se do jednoho s tím, že video grafu slouží jako překrytí v levém horním rohu.

Překrytí data s časem funguje stejně, akorát se musí vygenerovat každý snímek balíčkem Canvas. Dále se do videa přidá jako překrytí. Obě překrytí lze kombinovat dohromady se základním videem.

Všechny dočasné snímky a videa se ukládají do složek s unikátním identifikátorem, aby se předešlo konfliktům při souběžném zpracování více videí. To zamezuje konfliktů při souběžném zpracovávání více videí najednou. Po dokončení vytváření se všechny dočasné složky a soubory smažou.

Zobrazení časoběrných videí na klientské straně zajišťuje kontrolér, který má

na starosti několik úloh. Namísto odeslání celého videa najednou, se nastavuje hlavička HTTP Range, která umožní streamování videa po segmentech. Klientovi je odeslána odpověď s hlavičkami Content-Length a Content-Range, jež zahajuje streamování videa v dávkách, místo aby bylo nutné čekat na načtení celého souboru. Tento přístup zrychluje start videa, ale také efektivněji využívá síťové zdroje. Čtení a odesílání souboru na serveru se realizuje pomocí fs.createReadStream, což umožňuje práci s velkými soubory po částech bez nutnosti načítat celý obsah videa do paměti serveru. Tento přístup minimalizuje zátěž na serveru.

Jestliže si uživatel přeje video stáhnout místo jeho přehrání v prohlížeči, kontrolér toto dovoluje skrze požadavek. Díky nastavení hlavičky Content-Disposition dojde k spuštění události v prohlížeči, která nabídne video ke stažení.

Pokud dojde k nějaké chybě, nebo požadavek neobsahuje hlavičku Range, server se pokusí odeslat video celé. V případě většího problému server vrátí chybovou zprávu.

Kontrolér pro správu časosběrných videí zahrnuje také logiku pro získání všech videí, která zahrnuje stránkování. To umožňuje uživatelům procházení po stránkách a k vybrání videa, které se chce přehrát. Nakonec lze mazat jednotlivé videa a jejich záznamy z databáze pomocí identifikátoru.

### 3.1.7 Posílání emailu

Systém je schopný posílat jednou denně email ze serveru, informující o stavu rostliny.

Je potřeba nejdříve získat emailovou adresu od uživatele. To se dá provést, díky klientské části nebo API. Kromě emailu lze nastavit také čas, kdy má email přijít. Požadavek se validuje regulárním výrazem a poté se uloží do databáze.

Posílání emailu funguje na základě balíčku Nodemailer. S jeho pomocí se posílají emaily jednoduše různými způsoby, jako SMTP, přihlášení k emailu nebo i službami jako je Mailgun, což ho činí velmi univerzálním. Z finančních důvodů byl vytvořen pouze Google účet, jehož přihlašovací údaje jsou předávány Nodemaileru. Místo hesla se používá token, který se získá po nastavení účtu na vývojový. Tyto citlivé údaje jsou uloženy v proměnných prostředí.

Po vytvoření zprávy a přihlášení je možné emaily automaticky odesílat. Tento proces je automatizován plánovačem Cron v Node.js prostředí. Server si uchovává jeho instanci v proměnné a v případě, že uživatel změní čas nebo přihlašovací údaje, původní úloha je smazána a vytvoří se nová, aby se vše aktualizovalo.

### 3.1.8 Autentizace

Zabezpečení přístupu k serveru je zajištěno autentizačním systémem, který zabráňuje neautorizovanému přístupu k operacím, jako je prohlížení a úprava dat. Je potřeba se přihlásit pomocí přihlašovacích údajů. Jako výchozí přihlašovací údaje při úplném prvním spuštění systému jsou nastaveny na **admin** pro jméno i heslo. Údaje lze změnit v nastavení klientské části. V případě úspěšného přihlášení server odesílá nazpátek klientovi session cookie, která obsahuje identifikátor uživatele. Ta se

automaticky uloží v prohlížeči a zůstává aktivní do té doby, dokud uživatel prohlížeč nezavře, poté server vyžaduje opětovné přihlášení k ověření identity.

K zajištění bezpečnosti jsou session cookies nastaveny s atributem SameSite na hodnotu Lax, což pomáhá chránit v moderních prohlížečích proti útokům typu Cross-site Request Forgery tím, že musí cookie pocházet ze stejné domény, jinak se neuloží. Klient posílá cookie s každým požadavkem, což serveru umožňuje rozpoznat autorizovaného uživatele.










Pro RPI je zvolen jiný způsob - prostřednictvím REST API klíčů, aby nebyl problém s ukládáním session cookies. Tyto klíče se dají spravovat na klientské části, kde se mohou vygenerovat, zobrazit nebo smazat. API klíče umožňují RPI pouze nahrávat data ze senzorů společně s fotkou na server. V hlavičce musí být pokaždé klíč x-api-key. V databázi se uchovávají informace o klíčích, jako datum vytvoření a statistiky jejich využití.

### 3.1.9 Testování

Kontroléry, které obsahují hlavní logiku serveru, byly otestovány testovacím frameworkem Jest, jenž umožňuje tvorbu unit a integračních testů, pomocí jazyka Typescript. Tyto testy jsou vytvořeny k ověření funkčnosti, zajištění kvality a odhalení potenciálních chyb.

Testy pokrývají většinu funkcionalit, jako CRUD operace, práci se soubory, autentizaci a další. Pro simulaci interakce s databází a jinými službami je využíváno mockování, které umožňuje testování bez skutečného přístupu k těmto zdrojům.

Tabulka 3.1: Pokrytí testů. Zdroj: Vlastní zpracování.

Soubor	Zaznamenané řádky	Pokryté řádky	Chybějící řádky	Pokrytí %
apiKeyController.ts	46	37	9	 80.43%
authController.ts	49	45	4	 91.84%
emailSettingsController.ts	27	22	5	 81.48%
eventController.ts	16	14	2	 87.50%
plantSettingsController.ts	36	33	3	 91.67%
sensorDataController.ts	96	94	2	 97.92%
timelapsesController.ts	117	72	42	 61.54%
uploadController.ts	30	29	1	 96.67%
imageUtils.ts	48	43	5	 89.58%

Výsledky testů jsou automaticky a průběžně posílány na webové stránky Codecov při každém pushi do repozitáře prostřednictvím Github Actions. Codecov poskytuje detailní statistiky o pokrytí testů a přidává odznak do README projektu, který informuje o pokrytí testování.

## 3.2 Umělá inteligence

Samostatné měření senzorů není dostatečné pro posouzení zdraví rostliny. Rostlina může stále umřít nebo ji může postihnout nemoc. Z tohoto důvodu byl proveden pokus o vytvoření umělé inteligence, která by na základě snímku dokázala predikovat stav zdraví rostliny.

### 3.2.1 Detekce zdraví rostliny

Ke klasifikaci zdravotního stavu rostliny byla použita konvoluční neuronová síť, implementovaná v programovacím jazyce Python s využitím knihoven TensorFlow a Keras.

Model byl trénován na základě vlastních snímků, které byly pořízeny tímto systémem. Během experimentu prošla bazalka všemi stádii životního cyklu od zdravého stavu až po úplné odumření. Pro trénink modelu bylo nejprve nutné ručně rozdělit snímky do odpovídajících kategorií (zdravá, uschlá, mrtvá). Většina snímků byla vyhrazena pro trénování, zatímco další snímky sloužily k validaci během tvorby modelu a několik snímků bylo ponecháno následnému testování.

ImageDataGenerator rozšířil trénovací sadu o modifikované verze originálních obrázků rotací a transformací. Obrázky jsou rovněž převedené na stupně šedi a sníží se jejich rozlišení. Při trénování se používá váhování tříd vyrovnávající nerovnoměrné zastoupení tříd, to pomáhá modelu lépe se učit i přes menší počet trénovacích dat.

Model je sekvenční, což znamená, že je sestaven z vrstev navazujících na sebe, čímž vytváří síť, která projíždí obrázkem a hledá vlastnosti a vzory. Síť obsahuje konvoluční vrstvy pro extrakci klíčových vlastností, pooling vrstvy pro redukci objemu dat, flatten vrstvu pro transformaci dat do jednorozměrného vektoru a dense vrstvy, které provádějí klasifikaci. Využívají se aktivační funkce ReLU a Softmax, které určují, jakým způsobem se signály propagují skrze síť. Model je kompilován optimalizátorem Adam, který nastavuje váhy při trénování. [24].

Trénink probíhal s určeným počtem epoch, přičemž jako opatření byl nastaven mechanismus, který trénování předčasně ukončí v případě, že výsledky se nadále nezlepšují, což předcházelo přetrénování.

Model úspěšně dokáže rozpoznávat stav rostliny, nicméně jeho přesnost je omezená kvůli malému počtu snímků. Přesnost predikce je vysoká u bazalky, pro kterou byl původně trénován, ale pokud se model pokusíme použít na jiných typech rostlin nebo v značně odlišném prostředí, jeho výsledky nejsou spolehlivé. Je tedy potřeba rozšířit trénovací dataset o další snímky a typy rostlin, aby se zlepšila schopnost modelu.

### 3.2.2 Detekce nemoci rostliny

Vzhledem k omezenému množství dat se využil již existující model, který umožňuje detekovat nemoci rostlin.

Projekt je veřejně dostupný na platformě Github pod jménem Plant-Disease-Detection-and-Solution od autora Sahil Thakur. Detekce je založena na principu strojového učení s knihovnou Tensorflow podobně jako předchozí přístup, avšak využívá více vrstev. Výhodou je, že model je optimalizován pro menší velikost a není tak náročný na hardwarové zdroje. Celkově projekt slouží k rozpoznávání nemocí rostlin v mobilní aplikaci [25].

Pro trénování se používá PlantDoc Dataset, jenž obsahuje fotografie třinácti různých druhů rostlin. Snímky jsou již rozvrženy do složek podle specifických nemocí a jsou pořízené z různých prostředí a úhlů, což zaručuje robustnost modelu. Tento

dataset je populární a využívá ho mnoha projektů [26].

Tvorba modelů byla prováděna na webové platformě Google Colab, jež poskytuje vysoký výkon pro práci s Jupyter notebooky, které se hodí pro strojové učení. Umožňuje také napojení na vlastní Google disk. Tato služba zadarmo urychlila tvorbu modelu bez potřeby vlastního silného hardwaru.

Model zdraví		Model nemocí	
Tento model predikuje zdraví rostliny.		Tento model predikuje různé typy nemocí, které může rostlina mít.	
Název	Pravděpodobnost %	Název	Pravděpodobnost %
Zdravá	99.98%	Dýně - Prášková plíseň	99.94%
Uschlá	0.02%	Jahoda - Listové spálení	0.06%
Nemocná	0.00%	Jabloň - Strupovitost	0.00%
> SPUSTIT		> SPUSTIT	

Obrázek 3.3: Vyhodnocení stavu rostliny ve webové aplikaci. Zdroj: vlastní zpracování.

## 3.3 Klient

Klientská část je postavena na open-source knihovně React od známé společnosti Meta, dříve nazývaná jako Facebook. React je určen k tvorbě interaktivního a dynamického webového rozhraní. Produkčně ho používá mnoho populárních firem. K vývoji se používá také Typescript.

Hlavním principem je komponentní architektura, která umožňuje rozdělení uživatelského rozhraní do samostatných a znovupoužitelných souborů. Každá komponenta obsahuje vlastní logiku, stav, vstup a výstup.

### 3.3.1 Struktura klientského kódu

Kód je organizován do několika adresářů, které odrážejí běžné uspořádání projektu:

- **Components:** Obsahuje komponenty, které tvoří základní stavební bloky.
- **Pages:** Složka reprezentující stránky aplikace, využívá komponenty z **components** složky.
- **Locale:** Překlady textů ve formátu JSON pro podporu vícejazyčného obsahu.
- **Api:** Funkce pro komunikaci pomocí REST API.
- **Context:** Využívá React Context pro globální správu stavu aplikace. Usnadňuje sdílení dat mezi všemi komponentami.
- **Utils:** Pomocné funkce pro formátování.
- **Types:** Definice TypeScript typů pro zvýšení typové bezpečnosti.



### 3.3.2 Styl stránky

Namísto tradičních kaskádových stylů je nainstalován Tailwind CSS. Z největších výhod použití tohoto frameworku je možnost psát styly přímo do HTML kódu. To eliminuje potřebu vytvářet samostatné CSS soubory a tím značně urychluje proces vývoje.

Tailwind CSS nabízí všechny základní třídy a v případě potřeby je možné si rozšířit o vlastní. Všechny třídy jsou responzivní pro různé velikosti obrazovky, jako mobilní zařízení, tablety a stolní počítače.

Jako nadstavba pro Tailwind CSS je používán Daisy UI. Tato knihovna poskytuje předdefinované hotové komponenty, jako tlačítka, tabulky, karty a další. Obsahuje také využití barevných motivů, které definují vzhled celé stránky prostřednictvím tříd „primary“ a „secondary“, místo přímého určení barev. Díky této funkcionalitě bylo možné implementovat přepínání mezi světlým a tmavým režimem stránky.

### 3.3.3 Axios

K posílání požadavků na REST API se používá balíček Axios. Nejprve je vytvořena jeho instance s adresou serveru, což usnadňuje opakované volání.

V adresáři services se nachází funkce, které slouží pro práci s jednotlivými endpointy REST API serveru. Při práci s těmito funkcemi v komponentách se obvykle definují stavy pro načítání a případné chyby, které se pak vizuálně mohou prezentovat uživateli a informovat o průběhu požadavku.

V případě, že Axios zachytí stavový kód 401, kdy signalizuje problém s autentizací. Aplikace automaticky uživatele odhlásí a přesměruje ho na přihlašovací stránku.

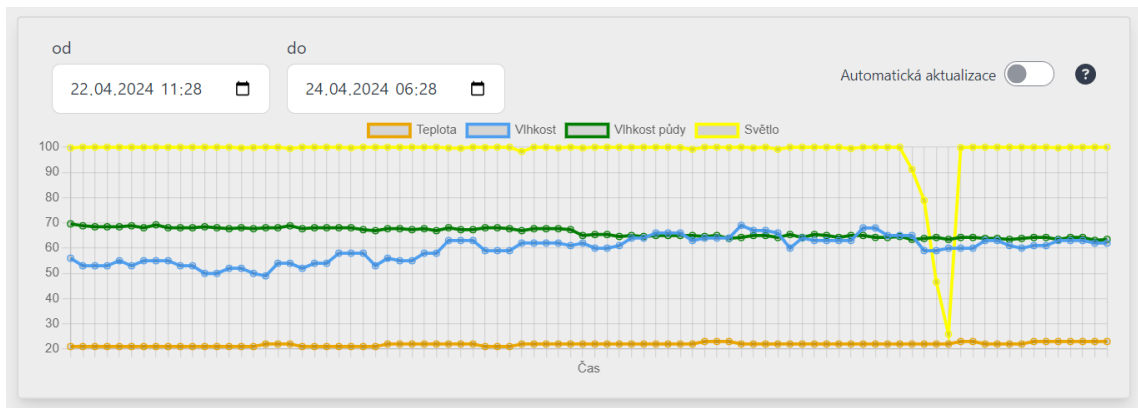
### 3.3.4 Lokalizace

Pro zaručení vícejazyčnosti klientské části je použita knihovna i18next a react-i18next, což umožňuje snadné přepínání mezi jazyky. Překlady jsou uloženy ve dvou JSON souborech, jeden pro češtinu a druhý pro angličtinu, kde se nachází klíče a jednotlivé překlady využívané v aplikaci.

Přepínání jazyků je umožněno tlačítkem, které zavolá `i18n.changeLanguage`, jenž obmění překladový soubor na požadovaný jazyk. Přístup k překladům v komponentách je zajištěn díky React hooku `useTranslation`. Tento hook poskytne funkci překladu, která se pak volá s klíčem v komponentách aplikace.

### 3.3.5 Graf

Vizualizace dat senzorů ve spojnicovém grafu je podobně jako na serverové části realizována balíčkem `chart.js`. Data jsou již předzpracována ze serveru, proto stačí pouze definovat na klientské části odpovídající popisky a barvy. Na grafu se standardně zobrazují všechny senzory najednou, ale uživatel si může dle svého vybrat, které senzory chce vidět.



Obrázek 3.4: Graf ve webové aplikaci. Zdroj: vlastní zpracování.

Komponenta grafu obsahuje také dvě vstupní pole pro filtraci časového rozsahu. Při přidání nových dat na server se graf automaticky aktualizuje. Toto nastavení lze, dle preferencí deaktivovat prostřednictvím přepínače.

### 3.3.6 React context

Ke správě globálního stavu v aplikaci je využít React context, který dokáže sdílet funkce a proměnné v celé aplikaci, bez nutnosti propojovat stav mezi jednotlivými komponentami.

Auth context je zodpovědný za autentizaci a autorizaci uživatelů. Obsahuje funkce pro přihlášení, odhlášení a ověření stavu. Tyto informace jsou následně propagované napříč celou aplikací. Používá se i pro ochranné trasy, které v případě nepřihlášeného uživatele přesměrovává zpět na přihlašovací stránku.

Server sent context sleduje, zda je uskutečněno otevřené připojení mezi serverovou a klientskou částí. Kontroluje přicházející signály a zda byl zaznamenán příchod nových dat. Poté vysílá signál všem komponentám o možnosti revalidace data, pro získání aktuálních informací.

## 4 Nasazení a zabezpečení

Vzhledem k omezenému výkonu RPI Zero 2 je serverová a klientská část aplikace nasazená na virtuálním serveru. Po jeho spuštění bylo potřeba provést několik klíčových kroků k správnému fungování a zabezpečení.

### 4.1 Konfigurace serveru

K bezpečnému připojení na virtuální server se namísto hesel nejdříve vytvořily nové SSH klíče na lokálním počítači. Klíče byly následně přidány do adresářů autorizovaných klíčů uživateli root a pro nově vytvořeného uživatele s omezenými právy. V konfiguračním souboru SSH bylo zakázáno přihlášení heslem pro roota. Autentizovat se může pouze pomocí klíče.

Kvůli prevenci proti útoku hrubou silou byl nainstalován fail2ban omezující přístup IP adres po opakovaně neúspěšném přihlášení. Doba blokace trvá jeden den.

Důležitou částí zabezpečení serveru hraje firewall, jenž kontroluje veškerý příchozí a odchozí provoz. Byl nastaven UFW umožňující jednoduchou správu síťového provozu bez nutnosti přímé manipulace s pravidly iptables.

### 4.2 Docker

Docker byl vybrán jako platforma sloužící k nasazení, díky své schopnosti izolovat jednotlivé aplikace nezávisle na operačním systému. Instalace byla provedena, dle oficiální dokumentace. Jako první kontejner byl spuštěn Portainer, který nabízí grafické rozhraní pro správu všech kontejnerů a celého Docker systému. Umožňuje řadu operací, jako je například sledování logů, spouštění, zastavování a také instalaci nových kontejnerů. Vzhledem k tomu, že Portainer zodpovídá za správu celé infrastruktury, je přístup chráněn silným heslem a byl omezen přístup pouze na rozsah IP adres Technické univerzity v Liberci.

### 4.3 Databáze

Jako náhrada za původní SQLite databázi a k plnému využití zdrojů virtuálního serveru byl nasazen MySQL kontejner. Bylo nutné nastavit silné heslo a omezit

přístup pouze pro síť univerzity, aby bylo možné provádět testování během vývoje. Serverová část komunikuje v produkčním prostředí s databází na lokální síti.

## 4.4 Webový server a šifrování

Webový server Nginx zpracovává požadavky klientů a směruje je na serverové aplikace. Pro zvýšení bezpečnosti mezi klientem a serverem stojí služba Cloudflare, která nabízí ochranu proti DDoS útokům, šifrování, správu DNS záznamu a mnoho dalšího. Na Cloudflare byla také zakoupena a registrována doména.

Komunikace mezi prohlížečem a Cloudflare i mezi Cloudflare a serverem je šifrovaná. Šifrování mezi prohlížečem a Cloudflare je zajištěno certifikáty SSL/TLS, které poskytuje skrz certifikační autoritu Let's Encrypt. Jsou zdarma a automaticky se obnovují. Na virtuálním serveru je potřeba mít veřejný certifikát, který umožňuje Cloudflare zašifrovat data posílaná k serveru, a soukromý klíč, který tato data dešifruje. Oba tyto soubory se musí odkazovat v konfiguraci Nginx [27].

## 4.5 Nasazení aplikace

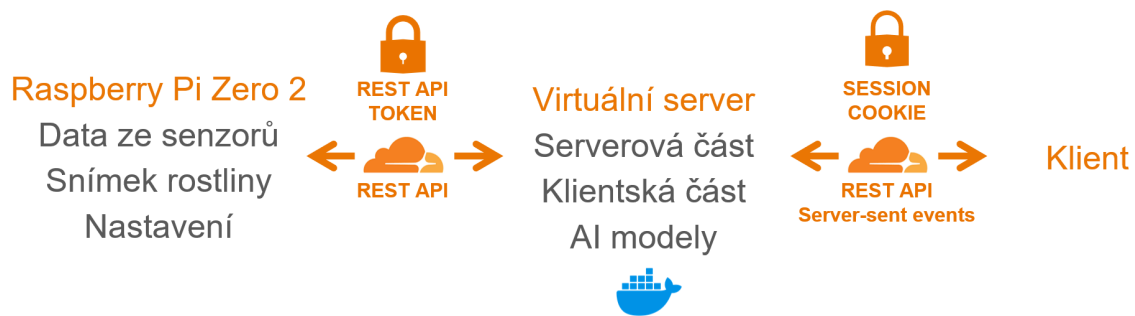
Serverová a klientská část mají oba svoje vlastní Docker soubory, s využitím dvoufázového sestavení, aby byla minimalizována velikost výsledných kontejnerů.

Klientská část v první fázi nastaví pracovní adresář uvnitř kontejneru, kam zkopíruje package.json, k nainstalování všech potřebných Node.js balíčků. Následně zkopíruje zbytek projektu a zahájí proces sestavení, jenž optimalizuje a zmenšuje kód projektu. V druhé fázi se do nového kontejneru kopíruje výsledek sestavení a spustí Nginx, který odkazuje na index.html.

V případě serverové části je postup podobný, ale ve své první fázi navíc provede migraci databáze pomocí Prisma. Ve druhé fázi instaluje další potřebné závislosti jako Python, Tensorflow a FFmpeg a spustí nakonec výslednou Node.js aplikaci.

Docker-compose spojuje tyto docker soubory dohromady. Umožňuje nasadit celý projekt najednou jediným příkazem. Navíc přidává Nginx kontejner, který funguje jako reverzní proxy, jenž sjednocuje serverovou a klientskou část pod jednu URL.

Celé nasazení na virtuální server bylo realizováno tak, že se projekt stáhnul z GitHubu, nastavily se potřebné proměnné prostředí a spustil se Docker-compose. Byl vytvořen vlastní Nginx konfigurační soubor a DNS záznam.



Obrázek 4.1: Schéma komunikace a nasazení projektu. Zdroj: vlastní zpracování.

## 5 Cenová kalkulace

Tabulka níže poskytuje cenový přehled všech hardwarových komponent potřebných pro realizaci projektu.

Tabulka 5.1: Cenový přehled produktů. Zdroj: Vlastní zpracování.

Produkt	Cena
Raspberry Pi Zero 2	600 Kč
Grove Base Hat	333 Kč
Kamera	695 Kč
Kamera - kabel	70 Kč
Grove - DHT senzor	123 Kč
Grove - relé	67 Kč
Grove - senzor světla	82 Kč
Grove - senzor vlhkosti půdy	164.6 Kč
Grove - displej	180 Kč
Čerpadlo	34 Kč
Hadička	29 Kč
LED světlo + kabel	173 Kč
Dřevěná konstrukce	210 Kč
Celkem	2760.6 Kč

Některé položky jako RPI, Grove Base Hat a Kamera s kabelem byly zapůjčené od vedoucí práce, kvůli tomu jsou skutečné náklady projektu nižší. Jako zdroj napájení byl využit starý kabel s adaptérem, který nebyl započítán do celkových nákladů. 3D tiskárna byla rovněž poskytnuta vedoucí práce, kde by se mohl započítat náklad za filament.

Virtuální server je díky studentskému statusu na rok zdarma. Cena za stejnou konfiguraci (2 GB ram, 60 GB SSD) je 380 Kč na měsíc. Doména slouží autorovi práce i pro jiné účely. V případě, že by byl zvolen výkonnější model RPI, nebylo by virtuálního serveru potřeba.

## 6 Testování

V této kapitole je popsáno, jak si systém vedl v pěstování rostlin a jak funguje webová aplikace z pohledu uživatelské interakce. Testování logiky serverové části naleznete v kapitole 3.1.9.

### 6.1 Systém

Během vývoje byly pro testovací účely zakoupeny bazalky v supermarketu, které jsou známe svou krátkou životností a jsou především určeny ke konzumaci.

V počátcích, systém disponoval pouze kamerou a DHT senzorem. Při pěstování bylo potřeba se soustředit na vizuální stav rostliny, kde bylo možné sledovat pohyb lístků a postupný vývoj v čase.

Postupně byly pořízené další senzory, což umožnilo přesněji určit potřeby rostliny a prostřednictvím senzoru vlhkosti půdy určit dobu zalévání. Nejdéle bazalka vydržela přes dva měsíce, následně z důvodu tvorby modelu umělé inteligence, byla úmyslně ponechána uschnout.

Během dalšího vývoje byl systém rozšířen o aktivní zavlažování a umělé LED osvětlení. Jako další rostlina byla zvolena jahoda, která je méně náročná na pěstování a byla vybrána pro ověření správnosti modelu pro detekci nemocí rostlin. I přes trénování modelu na bazalce dokáže v některých případech správně rozpoznat stav rostliny. Druhý model funguje podle očekávání. Na jahodě se objevily příznaky listového spálení, které model dokáže s vysokou pravděpodobností odhalit.

V nastavení klientské části je nyní možné přizpůsobit dobu zalévání, interval časosběru nebo zapnout funkci automatického zalévání. Rostlinu lze také zavlažovat manuálně tlačítkem v aplikaci, a v případě nepříznivého počasí nebo během noci zapnout umělé LED osvětlení. Po zapnutí LED světla rostlina reaguje vzpřímením svých listů, což dokazuje efektivitu osvětlení.

Celkově tento systém usnadňuje péči o rostliny, zlepšuje podmínky pro jejich růst a zvyšuje jejich životnost.

### 6.2 Webová aplikace

Webová aplikace je navržena s důrazem na intuitivní a přehledné uživatelské rozhraní. Různé funkcionality aplikace jsou doplněné o ikonky a vysvětlivky pro

lepší orientaci. Je dostupná jak v české, tak anglické lokalizaci a nabízí také tmavý režim pro pohodlnější používání.

Všechny formulářové prvky jsou otestovány. Zobrazují stav načítání a v případě chyby vypíší chybovou hlášku. Navigace mezi stránkami byla rovněž otestována. Tvorba časosběrných videí může trvat déle v závislosti na výkonu zařízení, množství snímků a možnosti překryvů.



## Závěr

V rámci bakalářské práce byl nejdříve proveden průzkum současných řešení v oblasti inteligentního pěstování rostlin, zahrnující jak komerční produkty, tak i otevřené a akademické projekty. Na základě této analýzy bylo navrženo vlastní hardwarové řešení založené na jednodeskovém počítači Raspberry Pi Zero 2, které disponuje řadou senzorů, kamerou, aktivním zavlažováním a LED osvětlením. Celý systém byl integrován do dřevěné konstrukce společně s 3D vytisknutými prvky.

Pro správu a vizualizaci získaných dat byla vyvinuta webová aplikace, která umožňuje navíc tvorbu časosběrných videí a přizpůsobení systému podle preferencí uživatele. Komunikace se serverovou částí probíhá skrze REST API a server-sent events, což umožňuje sledování dat v reálném čase.

Autentizace k serveru vyžaduje přihlašovací údaje pro přístup ke klientské části nebo REST API klíče k interakci s RPI. Z důvodu omezeného výkonu RPI je webová aplikace nasazená pomocí dockeru na virtuálním serveru, který byl důkladně konfigurován a zabezpečen na základě současných standardů.

Jako rozšíření projektu byl proveden pokus o vytvoření modelu k rozpoznání zdraví rostlin s využitím konvolučních neuronových sítí. Kromě vlastního modelu byl také přidán existující model pro rozpoznání nemocí rostlin. V budoucnosti by se tato práce mohla zaměřit na tuto oblast více. Další směr rozvoje je vytvoření vlastní obvodové desky a zdokonalení konstrukce, případně zvolení alternativního typu zavlažování.

Na základě testování finálního projektu lze potvrdit, že systém přináší značné výhody pro lepší růst rostlin a usnadňuje jejich pěstování.

## Použitá literatura

- [1] ALZA. *Jak na hydroponické pěstování rostlin* [online]. 2023. [cit. 2024-02-03]. Dostupné z: <https://www.alza.cz/jak-na-hydroponicke-pestovani-rostlin>.
- [2] LED ME GROW. *Aeroponické systémy* [online]. 2024. [cit. 2024-02-03]. Dostupné z: <https://www.ledmegrow.cz/aeroponicke-systemy/>.
- [3] SPECIALIZOVANÉ ZAHRADNICTVÍ. *Aeroponie* [online]. 2024. [cit. 2024-02-03]. Dostupné z: <https://www.specialnizahradnictvi.cz/aeroponie/>.
- [4] DOLTAK. *Kapková závlaha, jak na to?* [online]. [cit. 2024-04-11]. Dostupné z: <https://www.doltak.cz/blog-detail/kapkova-zavlaha-jak-na-to->.
- [5] DATART. *Chytré květináče* [online]. 2024. [cit. 2024-02-18]. Dostupné z: <https://www.datart.cz/novinky/produkty/chytre-kvetinace>.
- [6] ALZA. *Chytré květináče* [online]. 2024. [cit. 2024-02-18]. Dostupné z: <https://www.alza.cz/chytre-kvetinace/18862180.htm>.
- [7] AMAZON. *Masdio Ivy - Smart Flowerpots* [online]. 2024. [cit. 2024-02-07]. Dostupné z: <https://www.amazon.com/Masdio-Flowerpots-Intelligence-Temperature-Environment/dp/B0CPSY5HJY>.
- [8] CLICK AND GROW. *The Smart Garden 3* [online]. 2024. [cit. 2024-02-17]. Dostupné z: <https://www.clickandgrow.com/products/the-smart-garden-3>.
- [9] CLICK AND GROW. *Indoor Herb Gardens and Indoor Gardening Kits / Click and Grow* [online]. 2024. [cit. 2024-02-17]. Dostupné z: <https://www.clickandgrow.com/>.
- [10] HEUREKA. *Click and Grow SmartGarden CNG SG3 WHI Bílý* [online]. 2024. [cit. 2024-02-17]. Dostupné z: <https://kvetinace.heureka.cz/click-and-grow-smartgarden-cng-sg3-whi-bily/#specifikace>.
- [11] EMILOSTUFF. *Preventing Plant Death With Technology* [online]. 2020. [cit. 2024-02-22]. Dostupné z: <https://www.youtube.com/watch?v=9Fx9zQJe3H4>.
- [12] INSTRUCTABLES. *Fytó: Turn Your Plant Into Pet* [online]. 2024. [cit. 2024-02-22]. Dostupné z: <https://www.instructables.com/Fyt%C3%B3-Turn-Your-Plant-Into-Pet/>.
- [13] INSTRUCTABLES. *Raspberry Pi Powered IOT Garden* [online]. 2024. [cit. 2024-02-22]. Dostupné z: <https://www.instructables.com/Raspberry-Pi-Powered-IOT-Garden/>.

- [14] MATĚCHA, Tomáš. *Návrh systému pro automatické pěstování rostlin* [online]. Liberec, 2015 [cit. 2024-03-08]. Dostupné z: <https://dspace.tul.cz/server/api/core/bitstreams/1b8a94f5-4f30-4ebb-9b81-790f5a5a2217/content>. Diplomová práce. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií. Ing. Jiří Jelínek, Ph.D.
- [15] TRUHLAŘÍK, Václav. *Řízení automatického systému hydroponického pěstování rostlin* [online]. Praha, 2022 [cit. 2024-03-08]. Dostupné z: [https://dspace.cvut.cz/bitstream/handle/10467/101012/F3-BP-2022-Truhlarik-Vaclav-truhlarik\\_bakalarka\\_220520f.pdf?sequence=-1&isAllowed=y](https://dspace.cvut.cz/bitstream/handle/10467/101012/F3-BP-2022-Truhlarik-Vaclav-truhlarik_bakalarka_220520f.pdf?sequence=-1&isAllowed=y). Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická. Ing. Stanislav Vítek, Ph.D.
- [16] RPISHOP. *Raspberry Pi Zero 2 W s připájeným GPIO headerem* [online]. 2024. [cit. 2024-03-10]. Dostupné z: <https://rpishop.cz/raspberry-pi-zero/5114-raspberry-pi-zero-2-w-s-pripajenym-gpio-headerem.html>.
- [17] MONK, Simon. *Raspberry Pi Cookbook*. Second edition. Beijing: O'Reilly, 2016. ISBN 978-1-491-93910-9.
- [18] SEEED STUDIO. *Grove Base Hat for Raspberry Pi Zero* [online]. 2023. [cit. 2024-03-15]. Dostupné z: [https://wiki.seeedstudio.com/Grove\\_Base\\_Hat\\_for\\_Raspberry\\_Pi\\_Zero/](https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_Pi_Zero/).
- [19] BOTLAND. *Botland - obchod s robotikou* [online]. 2024. [cit. 2024-03-10]. Dostupné z: <https://botland.cz/>.
- [20] GUPTA, Devashish. *Capacitive vs. Resistive Soil Moisture Sensor* [online]. 2018. [cit. 2024-03-17]. Dostupné z: <https://www.hackster.io/devashish-gupta/capacitive-v-s-resistive-soil-moisture-sensor-e241f2>.
- [21] ALIEXPRESS. *STONEGO 28cm USB Cable with Switch ON/OFF* [online]. 2024. [cit. 2024-03-28]. Dostupné z: <https://www.aliexpress.com/item/1005005469373456.html>.
- [22] ALIEXPRESS. *USB 5V LED Grow Light* [online]. 2024. [cit. 2024-03-28]. Dostupné z: <https://www.aliexpress.com/item/1005005794691761.html>.
- [23] PRISMA. *Prisma: Simplify working and interacting with databases* [online]. 2024. [cit. 2024-03-24]. Dostupné z: <https://www.prisma.io/>.
- [24] KEITA, Zoumana. *Convolutional Neural Networks (CNN) with TensorFlow Tutorial* [online]. 2023. [cit. 2024-04-02]. Dostupné z: <https://www.datacamp.com/tutorial/cnn-tensorflow-python>.
- [25] DEVILSTUDIO27. *Plant Disease Detection and Solution* [online]. 2024. [cit. 2024-03-27]. Dostupné z: <https://github.com/DevilStudio27/Plant-Disease-Detection-and-Solution>.
- [26] KAYAL, Pratik. *PlantDoc-Dataset: A Dataset for Visual Plant Disease Detection* [online]. 2021. [cit. 2024-03-27]. Dostupné z: <https://github.com/pratikkayal/PlantDoc-Dataset>.

- [27] CLOUDFLARE. *What is Cloudflare?* [online]. 2024. [cit. 2024-04-10].  
Dostupné z: <https://www.cloudflare.com/learning/what-is-cloudflare/>.

## A Přílohy

- Zdrojové kódy jsou zveřejněné na platformě Github: <https://github.com/MarcelHor/bp-plant-project/tree/main>.
- Výsledky testování logiky serverové části jsou dostupné na: <https://app.codecov.io/gh/MarcelHor/bp-plant-project>.
- Webová aplikace je k dispozici na: <https://bp.marcel-horvath.com/>.
- Pro finální úpravu a korekturu textu byl použit ChatGPT verze 4.