

# Internetové technologie

Pink

Dokumentace

Barbora Hlubučková  
David Reichl  
Michal Špirk  
Jan Trejbal

LS 2020

# 1 O projektu

Toto je dokumentace ke studentskému projektu FAV z katedry Kybernetiky z předmětu Internetové technologie. Cílem bylo vytvořit internetovou aplikaci, která dokáže číst data ze senzorů, tyto data zpracovávat, odesílat na servery zákazníka, což byla společnost Aimtec, a dále zobrazit tyto hodnoty na webových stránkách, které poběží na virtuálním stroji na Západočeské univerzitě v Plzni.

Na projektu v našem týmu PINK se podíleli čtyři studenti a to:

- Michal Špírk
- David Reichl
- Jan Trejbal
- Barbora Hlubučková

Výsledek naší práce běžící na virtuálním stroji je možné najít pod adresou:

*http : //147.228.121.48 : 8881/*

## 2 Práce s virtuálním strojem

### 2.0.1 Vytvoření

Pro vytvoření virtuálního stroje je možné využít webových stránek stratus.zcu.cz, kde se vytvoří kopie z jedné ze 4 připravených šablon. Nejvíce nám vyhovovala šablona Debian, jelikož je to velice známý systém se kterým se dobře pracuje. Dále v záložce Access nalezneme přístup k příkazové řádce, kterou jsme delší dobu využívali jelikož to byl nejrychlejší přístup ke komunikaci na dálku a nebyla potřeba zadávání ip adres.

### 2.0.2 Komunikace

Při přístupu k CMD stroje doporučeno vytvořit jednotlivé uživatele a k nim určitá oprávnění, aby se vše neprovádělo přes “uživatele” root. Následně je potřeba zjistit veřejnou ip adresu virtuálního stroje. To nám zajistí příkaz ifconfig.

Dalším způsobem připojení, pro snadnější přenos souborů a složek mezi uživatelským počítačem a virtuálním strojem, bylo pomocí programu winSCP. Prostředí je formou oken, a nebo může mít vzhled jako, nejspíše každému známý, Total Commander. Takže uživatelské prostředí je velice přívětivé.

Posledním, námi využívaným komunikačním programem, byl Putty, kde stačí zadat ip adresu stroje a komunikace se naváže. Jedná se o prostředí příkazové řádky.

### 2.0.3 Zamýšlené pokusy

Využití práce s “cronem” a cron tabulkou, která by umožňovala automatické spuštění potřebných souborů v určitý čas, například každý den v 00:00. Dalo by se tím zamezit velkým výpadkům programů běžících na virtuálním stroji pro přenos souborů do firmy AIMTEC, nebo zajišťujících server a webové stránky. Maximální výpadek by potom byl maximálně 24 hodin.

Další myšlenka byla využití složek bin, nebo rc.local, do kterých by mělo stačit vložit soubory a ty se při každém startu spustí. To by mělo důsledek takový, že by nebylo vůbec zapotřebí hlídání běhu všech souborů. Bohužel to se rozběhnout nepodařilo.

### 2.0.4 Zprovoznění

Pro spuštění všech potřebných souborů a následného běhu všech souborů na pozadí může sloužit program Tmux. Příkazem apt-get install je potřeba Tmux nainstalovat do virtuálního stroje a následně pod názvem programu ho spustit. Práce zde je stejná jako v příkazové řádce. Jediné rozdíly jsou v tom že zde ještě funguje kombinace kláves “ctrl + b” následně kliknutí na určitou klávesu vykoná určitý úkon, jako je třeba nové okno, přejmenování okna, zrušení aktivního okna atd. Příkazem tmux list-commands se zobrazí veškeré možné příkazy a příkazem tmux list-keys se zobrazí klávesové zkratky.

### 2.0.5 Problémy

Z důvodů, že jsme nevyužívali anacondy/condy na virtuálním stroji, tak bylo občas obtížné najít názvy potřebných balíčků pro pip3, ale po nějaké době hledání se vždy podařilo názvy balíčků najít a nainstalovat. Dalším problémem bylo občasné zamrznutí virtuálního stroje, vše vždy vyřešil restart přes stratus.

## 3 Jak spustit a používat naší aplikaci

V této části bychom rádi popsali, jak spustit naší aplikaci v případě, že máte naše zdrojové soubory.

### 3.0.1 Potřebné soubory

Pro spuštění naší aplikaci je zapotřebí mít skripty client a server, tyto skripty jsou jádrem naší aplikace. V případě, že tyto skripty nemáme, není možné spustit naší aplikaci. Dále k přístupu k webové stránce jsou zapotřebí skripty HTML, kterých je 15. Nejdůležitějšími jsou html šablona2 a šablona-krize. Na první z nich je hlavní stránka s JavaScriptem na té druhé je šablona, která se ukáže pouze v případě problémů s daty. Dále je ke správné funkci zapotřebí několik json souborů s daty z předchozích dnů. Webové stránky je možné spustit i v případě absence jakýchkoliv dat. Na vliv našeho serveru to nemá vliv, jen nebude možné vidět data, která by zde byla v případě správné funkce našeho webu.

### 3.0.2 Jak a co spustit a na co si dát pozor

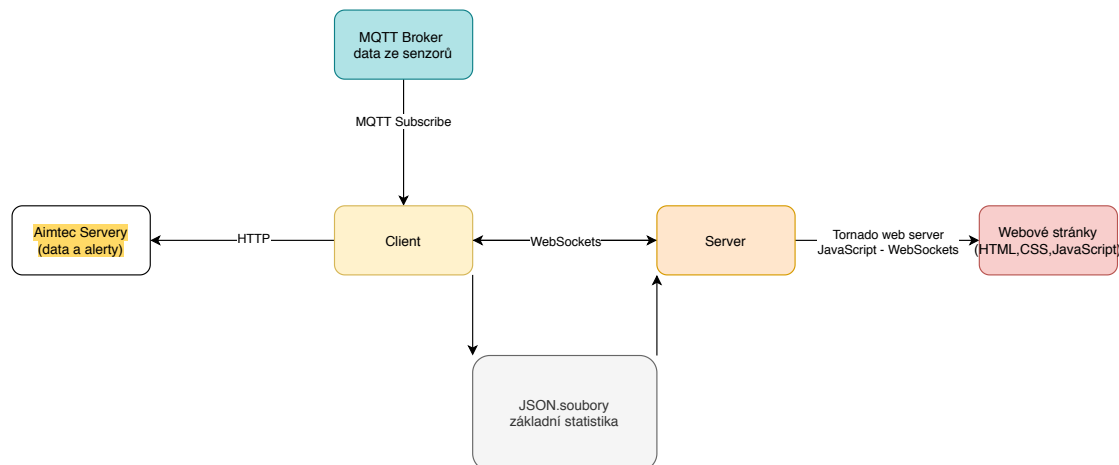
Pokud máme všechny potřebné soubory ke spuštění, je dobré začít spuštěním souboru server.py a poté až client.py. V případě, že spustíme nejprve klient, data, která přichází budou odesílány na servery zákazníka a budou ukládány do souborů, ale k serveru, který se ještě nezapl se nedostanou a zde budou ztracena, na naší webové stránce je nikdy neuvidíme. Důležité pro nás je, že nám aplikace nijak nepadá, ať ji spouštíme v jakémkoliv pořadí. Jakmile se tyto dvě aplikace spustí a začnou spolu komunikovat, je možné zobrazit webové stránky, které v případě, že nemají dostatek dat, napíší, že nemají data. To ale na funkčnost webových stránek nemá velkých vliv. V případě prvního spuštění je běžné, že aplikace nemá ještě nahromaděná data za minulé dny. Proto pro zobrazení dat z minulých dnů je potřeba počkat několik dní, dokud si je aplikace nenastřádá. Při zapínání a vypínání aplikace ojediněle může nastat chyba a to v případě, když se vypne server při zrovna probíhající komunikaci s klientem, v tomto případě spadne skript s klientem. Tato chyba se stává velmi ojediněle a objevila se hlavně při častém vypínání server.py.. Tato chyba je ale tak ojedinělá a vznikala pouze v tak speciálním případě, že běžného uživatele by neměla ovlivnit a na stálý běh aplikace nemá moc velký vliv.

### 3.0.3 Jak to spustit - ve 4 krocích

1. Mít soubory client.py a server.py
2. Mít všech 15 html stránek, pokud chci zobrazit webové stránky
3. Spustit server.py
4. Spustit client.py

## 4 Jak to funguje

### 4.1 Schéma aplikace



Obrázek 1: Schéma projektu

## 4.2 Skript Client

V naší aplikaci jsou dva hlavní skripty, které byly vytvořené v programovacím jazyku python. Tím prvním skriptem je client, který se stará hlavně o příjem dat, předzpracování dat a následné odesílání dat na servery Aimtecu. Tento skript také předává data našemu druhému skriptu a za každý den vytváří statistické soubory.

### 4.2.1 Poslouchání a přijímání dat

Příjem dat funguje za pomoci MQTT Brokeru, kdy skript za pomoci příkazu subscribe poslouchá data ze všech senzorů. Tyto data odpovídají formátu json nebo také slovníku v pythonu. Data jsme do slovníku přidali pouze v případě, že přišli ve správném formátu bez jakýchkoliv syntaktických chyb. Pokud data nesplňovaly tyto podmínky, tak byly označeny jako neplatné a dále se s nimi nepočítalo.

### 4.2.2 Odesílání dat a alertů na servery Aimtecu

Skript client se také staral o posílání dat na servery zákazníka. Pokud přišly data z našeho pink senzoru, tak jsme je obratem odesílali na server zákazníka za pomoci HTTP protokolu. Také jsme kontrolovali hodnoty a v případě, že se dostali mimo přípustné meze, což v našem případě bylo mimo hodnoty  $0 - 30^{\circ}C$ , došlo k vytvoření a odeslání alertu. Při překročení meze se tento alert odeslal pouze jednou. Nový alert se odeslal až v případě, kdy se hodnoty vrátili do normálních mezí a znovu je překročili. V případě, že při komunikaci nedostaneme odpověď 200, tak nastavíme hodnotu u Aimtec serverů jako offline.

### 4.2.3 Odesílání dat na náš skript server

Důležitou funkcionalitou je u skriptu Client předání dat skriptu Server. Kdyby toto předání se neuskutečnilo, nebylo by dost dobře možné zobrazovat data dynamicky v reálném čase. Toto předávání dat funguje za pomoci websocketu,

kdy se vytvoří komunikační kanál mezi skripty, předají se data a kanál se opět uzavře. Tato komunikace probíhá u Clienta asynchronně.

#### 4.2.4 Ukládání dat do souborů

Skript klient dále ukládá data ze senzorů do json souborů. Jedná se o dva typy souborů, jedním typem jsou všechna data pro každý senzor zvlášť, jsme tak schopni dohledat úplnou historii dat senzoru. Druhým typem souborů jsou jednotlivé dny, tyto soubory obsahují data o minimální, maximální a průměrné teplotě.

### 4.3 Skript Server

Jádrem serverového skriptu je framework Tornado, který jsme zvolili díky jeho dobré kompatibilitě s websockets a schopnosti rychle poskytovat služby velkému množství uživatelů.

#### 4.3.1 Příjem dat z klienta

Příjem dat z klienta probíhá pomocí websocket. Takto přijatá data jsou pokud jsou platná, jsou použita, společně s uloženými daty o poslední minimální, maximální a průměrné teplotě, k přepočítání těchto teplot a jejich opětovnému uložení do proměnných. Zároveň se do slovníku posledních došlých zpráv na pozici odpovídající barvě došlé zprávy uloží aktuální čas, který je následně využit při vykreslování stránky k zjištění, zda z daného senzoru došla data během posledních 100s, pokud ne, tak je stav senzoru nastaven na offline.

#### 4.3.2 Zobrazení online dat na webové stránce + javascript

Pro obsluhu uživatelské žádosti jsme využili sérii Handlerů. Výhodou našeho řešení je použití HTML šablon, které vykreslí webovou stránku pro předané senzory bez ohledu na jejich počet, tj. umožňuje velmi snadné škálování. Na pozadí běží Javascript, který zodpovídá za refreshování stránky v případě, že do serveru přijdou nová data. Data použitá pro vykreslení hlavní stránky jsou sestavena z globálních proměnných došlých pomocí websockets.

#### 4.3.3 Zobrazení dat z předchozích dnů

Pro zobrazení dat z předchozích dnů jsou použity soubory vytvářené klientem. Správný soubor se určí reference k dnešnímu datu. Soubory vytvářené klientem jsou relativně malé, protože obsahují pouze statistiku, tj. několik řádek informací. Správný soubor je načten, data jsou předzpracována a dosazena do šablon pro předchozí dny.

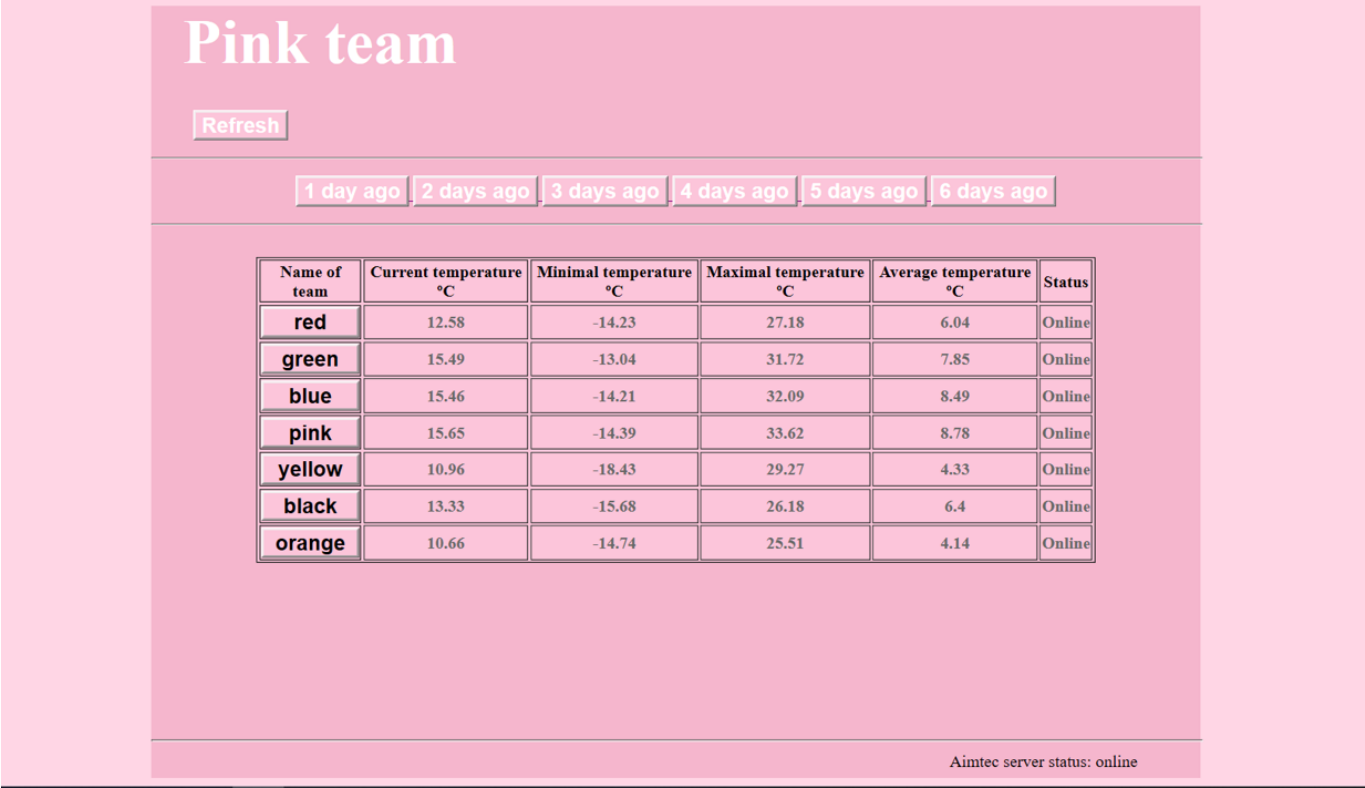
#### 4.3.4 Zobrazení dat pro každý senzor zvlášť

Pro zobrazení seznamu měření daného senzoru je využit soubor vytvářené klientem. Ten obsahuje soupis všech měření odpovídající dané barvě. Tato data jsou načtena předzpracována a dosazena do šablon pro jednotlivé barvy. Pokud je dané měření mimo přípustné meze povolených teplot, tak se zbarví do červena.

## 5 Vzhled stránky

Celkový vzhled stránky je ovlivněn tím, že jsme kladli důraz na možnost snadného škálování a použili jsme tedy tabulky. Dále jsme barvy uzpůsobili názvu našeho týmu. Kliknutím na barvu senzoru na hlavní stránce lze přejít na kompletní výpis měření. Kliknutím na tlačítka x day ago vykreslí data z předchozích dnů. Na hlavní stránce dále nalezneme informace o tom, zda dostáváme data z jednotlivých senzorů a zda se nám daří odesílat data na servery Aimtecu.

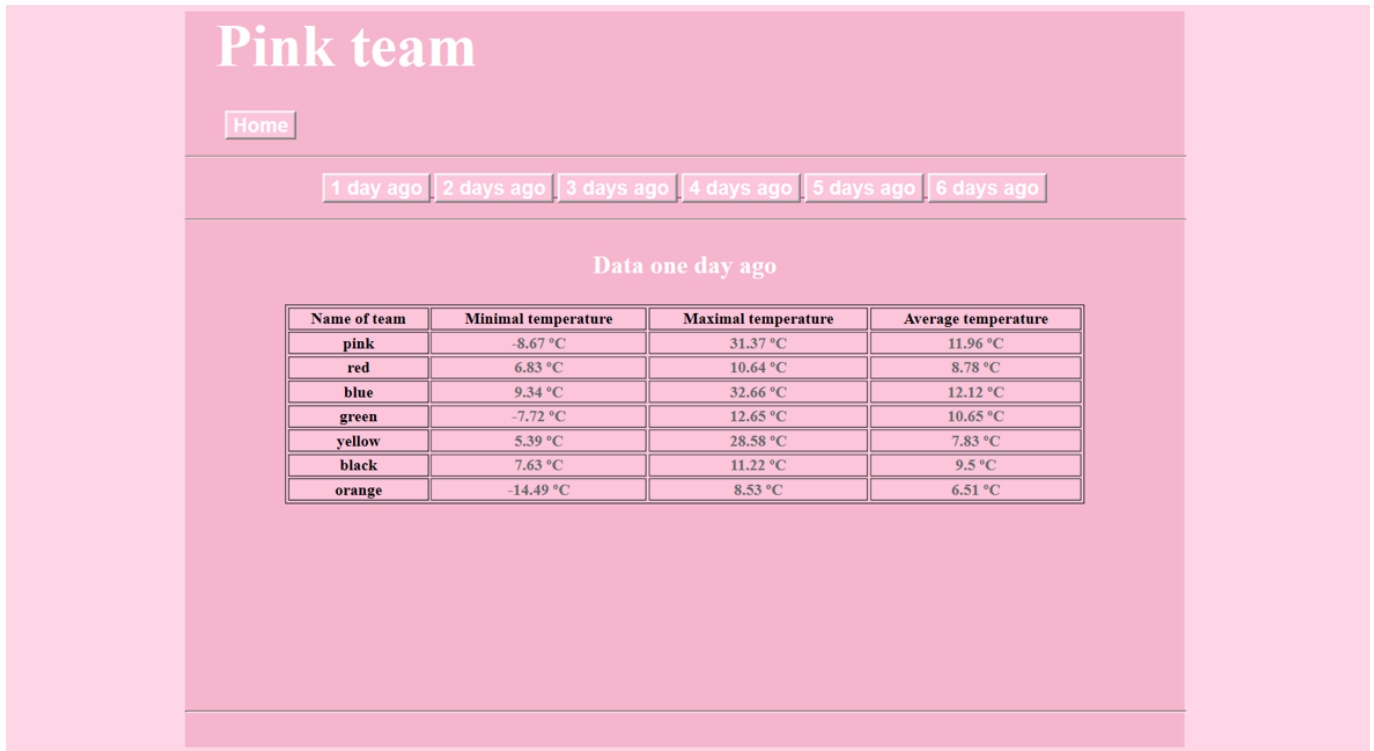
Obrázek 2: Obrázek webové stránky



Obrázek 3: Obrázek stránky jednotlivého senzoru



Obrázek 4: Obrázek stránky statistiky předchozího dne



## 6 Závěr

Naše řešení vyhovuje požadovaným parametrům zadání. Hlavní stránka našeho webu ukazuje všechny požadované hodnoty, je přehledná a uživatele nepřehlcuje zbytečně velkým množstvím informací. Další informace najde uživatel na vedlejších stránkách, na které lze snadno překliknout. Naše řešení by šlo rozšířit o grafy naměřených hodnot a případně stránky s komplexnější statistikou měření (např. medián měření, počet naměřených teplot v intervalech apod.). Během projektu jsme získali zkušenosti s velkým množstvím technologií a vyzkoušeli jsme si práci s návody a dokumentací při učení se novým technologiím. Velkou výzvou a zajímavou zkušeností byla práce na projektu bez osobního kontaktu a používání technologií pro komunikaci ve skupině. Věříme, že nám tato zkušenost pomůže při dalších skupinových i individuálních projektech.