|  |  |  |
| --- | --- | --- |
| **Dokumentacja Projektu** | | **JPiTI** |
| **Autor** | Sebastian Kaput, 106577 | **Data oddania** |
| **Kierunek, rok** | Mechatronika, III rok, st. Stacjonarne (3,5-l) | 15.07.2022 |
| **Specjalizacja** | - |
| **Grupa** | LAB 1 | **Ocena** |
| **Temat projektu** | Formularz zakupowy wraz z panelem administracyjnym |  |

# Opis tematyki projektu:

Projekt polega na wykonaniu w pełni działającego formularza zakupowego, który po odpowiednim jego uzupełnieniu i przejściu walidacji, zapisuje dane w chmurze na serwerze MongoDB.

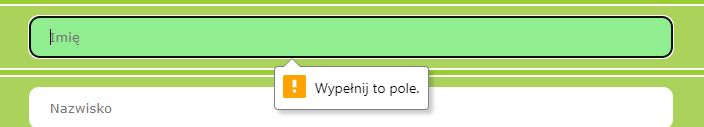
Obraz zawierający tekst

Opis wygenerowany automatycznie

Użytkownik posiada możliwość uzupełnienia formularza oraz złożenia zamówienia przyciskiem „Kup Teraz”.



Po jego naciśnięciu zostanie on poinformowany o tym, czy dane zostały prawidłowo wpisane, gdy wystąpi błąd, formularz wskaże, gdzie został popełniony.



W przypadku prawidłowego wypełnienia wyskoczy komunikat o pozytywnym złożeniu zamówienia.

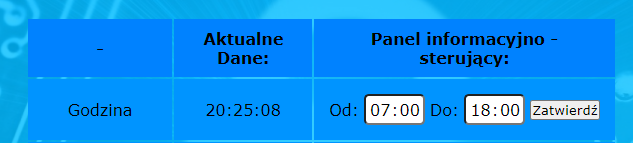
Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po stronie administratora istnieje możliwość ustawienia godzin funkcjonowania sklepu.

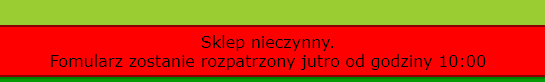


Po ustawieniu zakresu godzin, użytkownik będzie informowany o tym, czy sklep jest otwarty, czy zamknięty. W przypadku, gdy sklep będzie otwarty, będzie wyświetlana godzina, do której pozostanie on czynny.

Obraz zawierający tekst, żółty, urządzenie

Opis wygenerowany automatycznie

W innym razie pojawi się informacja o jego zamknięciu oraz kiedy zostanie ponownie otworzony.



Następnie administrator ma możliwość ustawienia ilości sztuk produktu dostępnych na magazynie.



Może on wpisać zakres jedynie od 0 wzwyż. Próba wpisania innej wartości będzie skutkowała pojawieniem się komunikatu o nieprawidłowym działaniu.

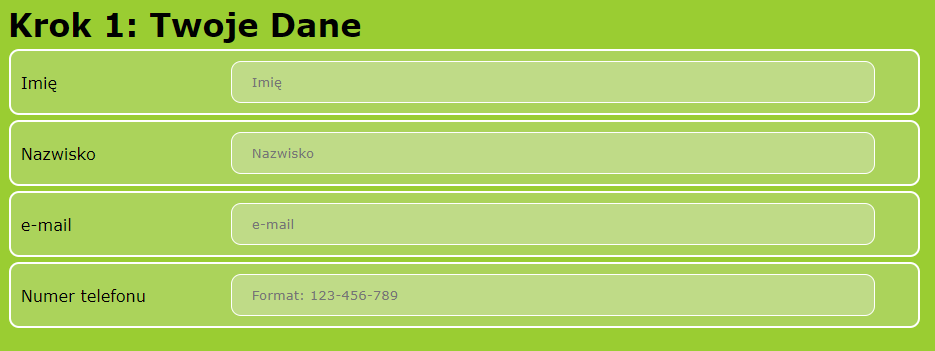
Obraz zawierający tekst

Opis wygenerowany automatycznie

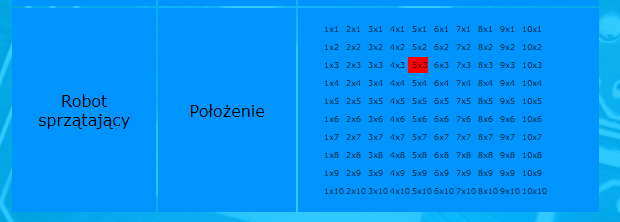
Gdy stan na magazynie spadnie do wartości „0”, formularz zostanie zablokowany i nie będzie można składać więcej zamówień, a pola do wpisywania danych zostaną wyłączone.

Poniżej przykład formularza aktywnego, gdy stan na magazynie >0 oraz gdy ilość sztuk spadła do 0.





Magazyn wyposażony jest w robota sprzątającego który udostępnia swoje położenie z pokoju o wymiarach 10 x 10 metrów. Dane te są przetwarzane i wyświetlane w tabeli 10 x 10 na panelu administratora z aktualnym jego położeniem, gdyż pracownicy często potykali się o niego nie wiedząc gdzie się znajduje.



Dodatkowo administrator może podejrzeć aktualne zamówienia wyświetlone w tabeli wraz ze wszystkimi informacjami wpisanymi przez użytkownika. Wygenerowanie tabeli następuje po kliknięciu przycisku „Zamówienia”



2) W dokumentacji mają być zawarte informacje w postaci opisów bądź wypunktowań: • opis tematyki projektu – opis słowny założeń projektu, jaki fragment rzeczywistości jest objęty działaniem aplikacji, co użytkownik może zrobić w aplikacji,

# Opis wykorzystanych technologii

Do wykonania projektu wykorzystano technologie programowania takie jak:

- HTML

- CSS

- JS

- EJS

W celu działania aplikacji, wymagane są paczki takie jak:

-ejs (<https://ejs.co/>)

-espress (<https://developer.mozilla.org/pl/docs/Learn/Server-side/Express_Nodejs/Introduction>)

-mongoose (<https://www.mongodb.com/>)

-body-parser (<https://www.npmjs.com/package/body-parser>)

Obraz zawierający tekst

Opis wygenerowany automatycznie

Do wykonania projektu wykorzystani środowisko takie jak Visual Studio Code (dostępne pod adresem https://code.visualstudio.com/ )

Strona została wykonana w HTML / EJS, za ustawienia graficzne, responsywność, zachowanie elementów na stronie odpowiadał CSS, natomiast za funkcjonowanie i wykonywanie akcji np. po kliknięciu przycisku stał JavaScript. Całość została umieszczona na serwerze wykonanym w NodeJS.

# Instrukcja uruchomienia aplikacji & co trzeba mieć zainstalowane

Należy uruchomić główny folder z plikami w Visual Studio Code, dokonać instalacji wymaganych paczek „NPM Install”, dokona się wtedy instalacja wszystkich wymaganych plików używanych w programie, a następnie uruchomić w terminalu „NPM Start”

- body-parser (wersja1.20.0)

- ejs (wersja 3.1.8)

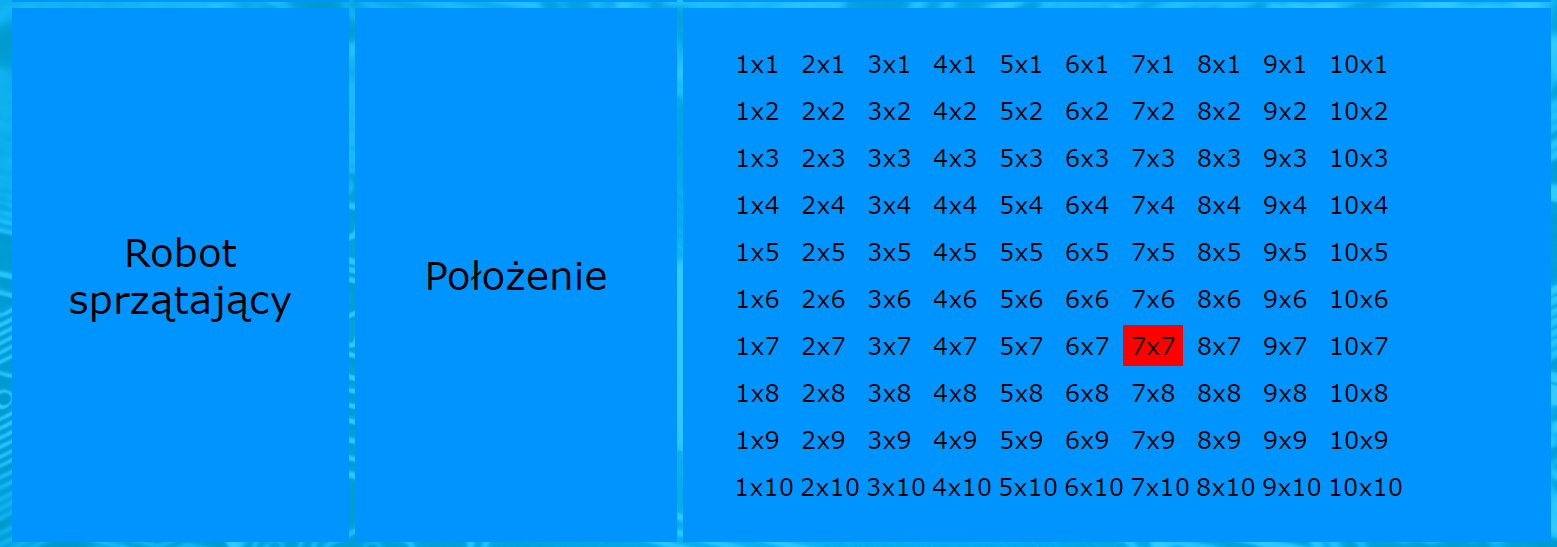
- express (wersja 4.18.1)

- mongoose (wersja 6.4.0)

- nodemon (wersja 2.0.16)

# Fragmenty kodu:

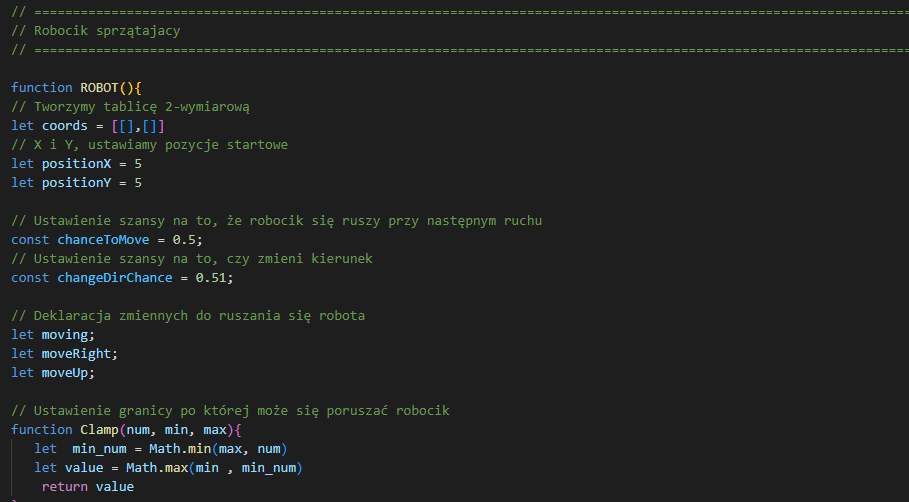
Tabela wyświetlająca położenie robota, kolorująca backgroundColor pola na czerwono:



Zdefiniowanie funkcji „ROBOT”, deklarujemy tablicę 2-wymiarową, ustawiamy pozycję startową 5,5

Ustawiamy const chanceToMove na 0.5 (czyli 50% szans na to, że robot wykona ruch)

Ustawiamy changeDirChance na 0.51 ( mamy 51% szans na to, że robot zmieni kierunek)



Ruch robota po osi X oraz Y

Obraz zawierający tekst

Opis wygenerowany automatycznie

Losowanie random liczby do wykonania ruchu, kolorowanie przebytej trasy.  
Ustawienie ruchu co 1000ms (1s)

Obraz zawierający tekst

Opis wygenerowany automatycznie

Ustawienie ilości sztuk sprzętu dostępnego na magazynie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Ustawienie zegarka zasysającego obecny czas z systemu. Następnie tworzymy pole do ręcznego ustawiania zakresu w jakim nasz sklep z internetowym formularzem jest aktywny.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Zapisujemy w pamięci naszą godzinę którą ustawiliśmy powyżej, następnie przyrównujemy ją do czasu systemowego i wyświetlamy odpowiednie dla danych godzin komunikaty.

Obraz zawierający tekst

Opis wygenerowany automatycznie