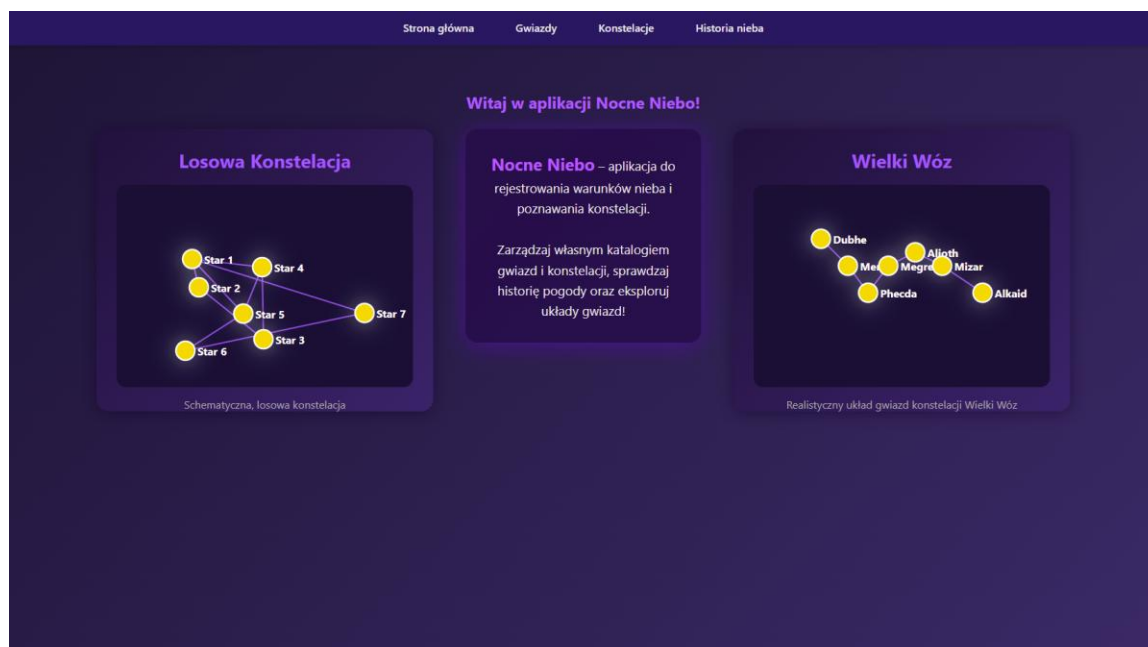


Nocne Niebo – System obserwacji gwiazd i konstelacji

1. Opis aplikacji

Aplikacja webowa służąca do zarządzania obserwacjami gwiazd, konstelacji oraz warunków pogodowych. Pozwala dodawać, edytować, przeglądać i usuwać gwiazdy oraz konstelacje, przypisywać gwiazdy do konstelacji, a także prowadzić dziennik warunków nieba. System wspomaga amatorskich astronomów oraz nauczycieli w dokumentowaniu obserwacji nieba. Umożliwia prowadzenie katalogu gwiazd i konstelacji, rejestrowanie warunków pogodowych każdej nocy oraz szybkie wyszukiwanie i prezentowanie informacji.



2. User Stories (Przykładowe scenariusze)

1. Jako użytkownik chcę móc dodać nową gwiazdę z nazwą, opisem i zdjęciem, aby katalogować własne obserwacje.
2. Jako użytkownik chcę przypisać gwiazdę do konstelacji, aby odwzorować układy gwiazd na niebie.
3. Jako użytkownik chcę przejrzeć szczegóły dowolnej konstelacji, aby zobaczyć, z jakich gwiazd się składa.
4. Jako użytkownik chcę rejestrować warunki nieba (pogoda, faza księżyca, mgła), aby wiedzieć, kiedy są najlepsze warunki do obserwacji.

5. Jako nauczyciel chcę zaświecać i wygaszać gwiazdy lub konstelacje, aby pokazywać ich widoczność w danej nocy.
6. Jako użytkownik chcę szybko odnaleźć wybraną gwiazdę lub konstelację i zobaczyć jej szczegóły oraz zdjęcie.

3. Odbiorcy systemu

- Amatorzy astronomii (osoby hobbystycznie obserwujące niebo)
- Nauczyciele i uczniowie (wykorzystujący aplikację do edukacji o astronomii)

4. Potencjalne korzyści dla użytkowników

- Organizacja własnych obserwacji astronomicznych
- Szybki dostęp do katalogu gwiazd i konstelacji
- Możliwość dzielenia się obserwacjami podczas zajęć lub spotkań kółek naukowych

5. Instrukcja uruchomienia

Wymagania: Node.js 18+, npm

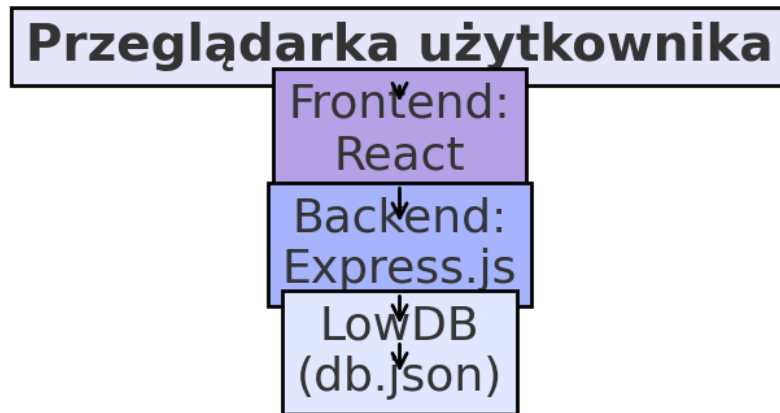
Backend (Node.js + Express + LowDB):

```
cd backend  
npm install  
npm start (http://localhost:3001/)
```

Frontend (React):

```
cd frontend  
npm install  
npm start (http://localhost:3000/)
```

6. Diagram architektury



Schemat komunikacji: Przeglądarka → Frontend (React) → REST API (Express) → LowDB (db.json)

7. Przykładowe endpointy API

Metoda	Endpoint	Opis	Przykład
GET	/api/stars	Pobierz listę wszystkich gwiazd	GET http://localhost:3001/api/stars
POST	/api/stars	Dodaj nową gwiazdę	POST .../stars { name, description, image }
PUT	/api/stars/:id	Edytuj dane gwiazdy	PUT .../stars/123 { ... }
DELETE	/api/stars/:id	Usuń gwiazdę	DELETE .../stars/123
GET	/api/constellations	Pobierz wszystkie konstelacje	GET .../constellations
POST	/api/constellations	Dodaj konstelację	POST .../constellations { name, ... }

POST	/api/constellations/:cid/add-star/:sid	Przypisz gwiazdę do konstelacji	POST .../constellations/1/add-star/2
GET	/api/sky-conditions	Pobierz historię warunków nieba	GET .../sky-conditions

Pełna lista endpointów w kodzie backendu. Każdy endpoint obsługuje poprawne statusy HTTP (200, 201, 404, 400).

Przykładowe zapytanie (curl):

```
curl -X POST http://localhost:3001/api/stars -H "Content-Type: application/json" -d '{"name": "Sirius", "description": "Najjaśniejsza gwiazda", "image": "https://..."}'
```

8. Przykładowe dane (db.json)

```
{
  "stars": [
    { "id": "1", "name": "Sirius", "description": "Najjaśniejsza gwiazda",
      "image": "https://..." }
  ],
  "constellations": [
    { "id": "1", "name": "Orion", "description": "Rozpoznawalna konstelacja",
      "image": "https://...", "stars": ["1"] }
  ],
  "skyConditions": []
}
```

9. Statusy odpowiedzi HTTP

- 200 – OK (pobranie, edycja, usunięcie)
- 201 – Created (dodanie)
- 400 – Błędne dane wejściowe
- 404 – Nie znaleziono zasobu
- 500 – Błąd serwera

10. User Experience – ważne element

- **Responsywny design** (przynajmniej strona Gwiazdy na desktop/mobile)
- **Powiadomienia o błędach/sukcesie**
- **Wybór daty i godziny w Historii Nieba**
- **Walidacja pól na frontendzie i backendzie**

11. Testy backend

Testy znajdują się w backend/backend.test.js

1. cd backend
2. npm test

Po uruchomieniu testów – Baza (db.json) zostanie zmodyfikowana! Jeśli nie chcemy, by testy mieszały w Naszych danych, warto podmienić plik db lub zresetować bazę przed/po testach

12. Testy frontend

Testy znajdują się w frontend/src/Stars.test.js

1. cd frontend
2. npm test

13. Autorzy projektu

Leon Stolecki 146031

Kacper Dombrowicz 145994

Patryk Przydanek 146025