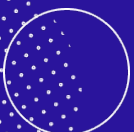
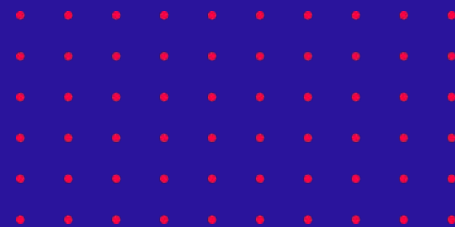




# ИТОГОВЫЙ ПРОЕКТ

---

Программирование  
на Typescript



## ЦЕЛЬ РАБОТЫ

Создать приложение для отображения погоды в выбранных городах на React с использованием TypeScript, а также библиотеки компонентов React Bootstrap и препроцессора SCSS.

## Что нужно сделать

- ☒ **01** Создать новое приложение на основе Create React App или NextJS с Typescript, установить React Bootstrap и SCSS

- ☐ **02** Создать 2 страницы:

### 2.1 Главная страница

- Отображает заголовок (рис. 1, 2, см в приложении) карточки с текущей погодой для 6 городов: 'Moscow', 'St Petersburg', 'Rostov-on-Don', 'Vladivostok', 'Krasnodar', 'Yekaterinburg'.
- Карточки должны отображаться адаптивно размеру экрана, используя встроенный Grid от React Bootstrap. На широких экранах 3 карточки в строку, затем 2 на более узких и 1 карточка в строке на мобильных размерах экранов.
- На карточке должны выводиться:
  - заголовок с названием города
  - текущая температура
  - общее состояние погоды (ясно, переменная облачность, пасмурно)
  - скорость ветра в м/с
  - кнопка "Смотреть прогноз", которая ведет на страницу с прогнозом погоды на ближайшие 7 дней

- Каждая карточка должна сделать 2 запроса на API:
  - Поиск координат города по его названию (название города карточка получает через props, используйте англоязычные названия, выделенные выше)
   
[https://geocoding-api.open-meteo.com/v1/search?name=\\${city}](https://geocoding-api.open-meteo.com/v1/search?name=${city})
  - Получение текущего прогноза по координатам
   
[https://api.open-meteo.com/v1/forecast?latitude=\\${lat}&longitude=\\${lng}&current\\_weather=true&windspeed\\_unit=ms](https://api.open-meteo.com/v1/forecast?latitude=${lat}&longitude=${lng}&current_weather=true&windspeed_unit=ms)

## 2.2 Страница прогноза погоды

URL страницы формируется следующим образом:

- <http://localhost:3000/weather/St%20Petersburg?lat=59.93863&lng=30.31413>, где значения выделенные синим передаются в ссылке с главной страницы (пробел в названии St Petersburg заменяется на спецсимвол %20 автоматически браузером).
- На странице расположена ссылка "назад", ведущая на главную страницу (<http://localhost:3000/>)
- Заголовок страницы (рис. 3, 4, см в приложении) и таблица с прогнозом погоды на 7 дней, состоящее из даты в формате (15/06, 16/06), минимальной прогнозируемой температуры в выбранную дату
- Максимальной прогнозируемой температуры в выбранную дату, прогнозируемой погоды (ясно, переменная облачность, пасмурно и т.д.)
- Для получения данных страницы прогноза погоды использовать следующий эндпоинт
   
[https://api.open-meteo.com/v1/forecast?latitude=\\${lat}&longitude=\\${lng}&daily=weathercode,temperature\\_2m\\_max,temperature\\_2m\\_min&timezone=GMT](https://api.open-meteo.com/v1/forecast?latitude=${lat}&longitude=${lng}&daily=weathercode,temperature_2m_max,temperature_2m_min&timezone=GMT)

- ☐ **03** Использовать готовые компоненты из React Bootstrap, такие как Card, Table, Container, Row, Col, Button, Spinner для прелоадера. В особых случаях можно использовать свои стили в SCSS- модулях или глобальных файлах SCSS
- ☐ **04** При получении данных с API используйте прелоадер (рис. 5 см в приложении)
- ★ ☐ Создайте компонент AutocompleteInput, который, используя API [https://geocoding-api.open-meteo.com/v1/search?name=\\${ev.target.value}](https://geocoding-api.open-meteo.com/v1/search?name=${ev.target.value}) автодополняет название города, вводимого в текстовом поле
- При клике по выпадающему варианту Input заполняется полным названием города, выпадающий список скрывается и происходит переадресация на страницу с прогнозом погоды в данном городе. Компонент должен использовать <Form.Control> и <ListGroup> из React bootstrap
  - Дизайн элемента и расположение на главной странице см на рисунках 6-8 (см в приложении)



## Рекомендации для выполнения

- 01** Данные о погоде взять с API <https://open-meteo.com/>
- 02** Гуглите и изучайте документацию, так как это самый ценный навык для разработчика!
- 03** Переиспользуйте компоненты и хуки, где это возможно, например карточки, заголовки, хуки получения данных и управления состоянием (data, loading, error) можно поместить в Custom hooks
- 04** Для расшифровки weather code можно использовать подготовленный мной на основе документации API словарь

```
const weatherInterpretationRu: WeatherInterpretation = {
  0: "Ясное небо",
  1: "Преимущественно ясно",
  2: "Переменная облачность",
  3: "Пасмурно",
  4: "Облачно",
  5: "Очень облачно",
  6: "Пасмурно",
  45: "Туман и мгла",
  48: "Туман и мгла",
  51: "Морось: легкая интенсивность",
  53: "Морось: умеренная интенсивность",
  55: "Морось: сильная интенсивность",
  56: "Ледяная морось: легкая интенсивность",
  57: "Ледяная морось: сильная интенсивность",
  61: "Дождь: слабая интенсивность",
  63: "Дождь: умеренная интенсивность",
  65: "Дождь: сильная интенсивность",
  66: "Ледяной дождь: легкая интенсивность",
  67: "Ледяной дождь: сильная интенсивность",
  71: "Снег: слабая интенсивность",
```



```

73: "Снег: умеренная интенсивность",
75: "Снег: сильная интенсивность",
77: "Снежные зерна",
80: "Дождь с прояснениями: слабая интенсивность",
81: "Дождь с прояснениями: умеренная интенсивность",
82: "Дождь с прояснениями: сильная интенсивность",
85: "Снег с прояснениями: слабая интенсивность",
86: "Снег с прояснениями: сильная интенсивность",
95: "Гроза: слабая или умеренная",
96: "Гроза с мелкой градиной",
99: "Гроза с крупной градиной"

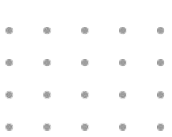
```

- 05** Форматирование дат и конвертирование WeatherCode, полученного с API проводите в отдельных функциях, которые положите в папку utils/
- 06** Не забывайте типизировать компоненты, хуки и функции
- 07** Получайте данные с помощью стандартного метода fetch
- 08** Для удобства работы с автодополнением полей данных с API напишите хотя бы частичную типизацию для данных с сервера
- 09** Для работы с scss поставьте пакет sass из npm
- 10** Используйте массив с названиями городов для отрисовки карточек

```

const cities = [
  'Moscow',
  'St Petersburg',
  'Rostov-on-Don',
  'Vladivostok',
  'Krasnodar',
  'Yekaterinburg'
]

```



## Критерии оценки

- |           |  |         |
|-----------|--|---------|
| <b>01</b> | Создана структура проекта, установлен Create react app или NextJS с Typescript, React bootstrap и SCSS                                   | 1 балл  |
| <b>02</b> | Написана главная страница, подключены данные с API (запрос поиска координат по имени города, затем запрос текущей погоды по координатам) | 2 балла |
| <b>03</b> | Написана страница деталей погоды со ссылкой на главную, заголовком и таблицей, подключены соответствующие данные с API                   | 2 балла |
| <b>04</b> | Переиспользуются компоненты, функции и хуки  | 2 балла |
| <b>05</b> | Типизированы все компоненты, функции и хуки  | 2 балла |
| <b>06</b> | Отсутствуют ошибки приложения в рантайме (красные консоль-логи)  | 1 балл  |
| <b>07</b> | Выполнено задание со звездочкой  | 3 балла |

**Максимальное** количество баллов

**13 баллов**

**Минимальное** количество баллов, чтобы преподаватель принял работу

**7 балла**

# Приложение

Рисунок 1. Десктопный вид главной страницы

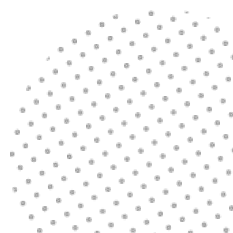
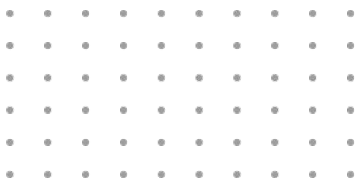
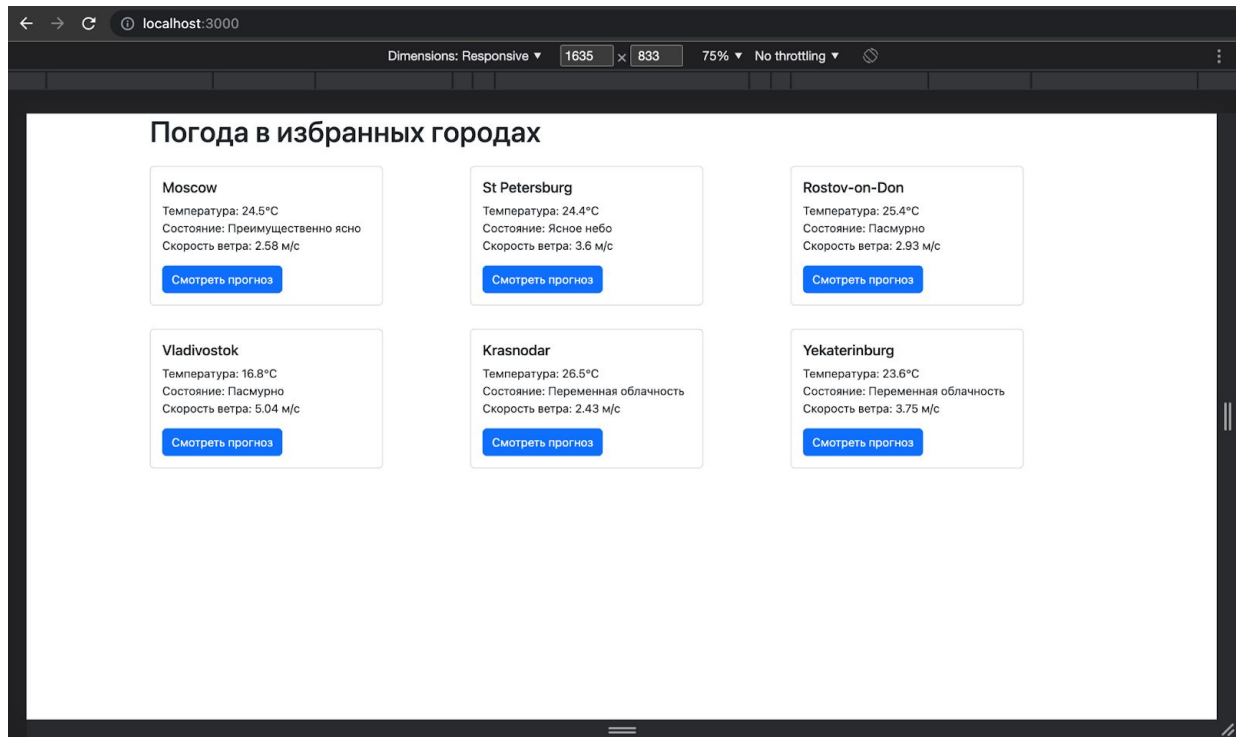




Рисунок 2. Мобильный вид главной страницы

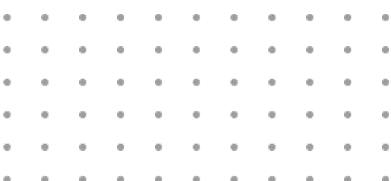
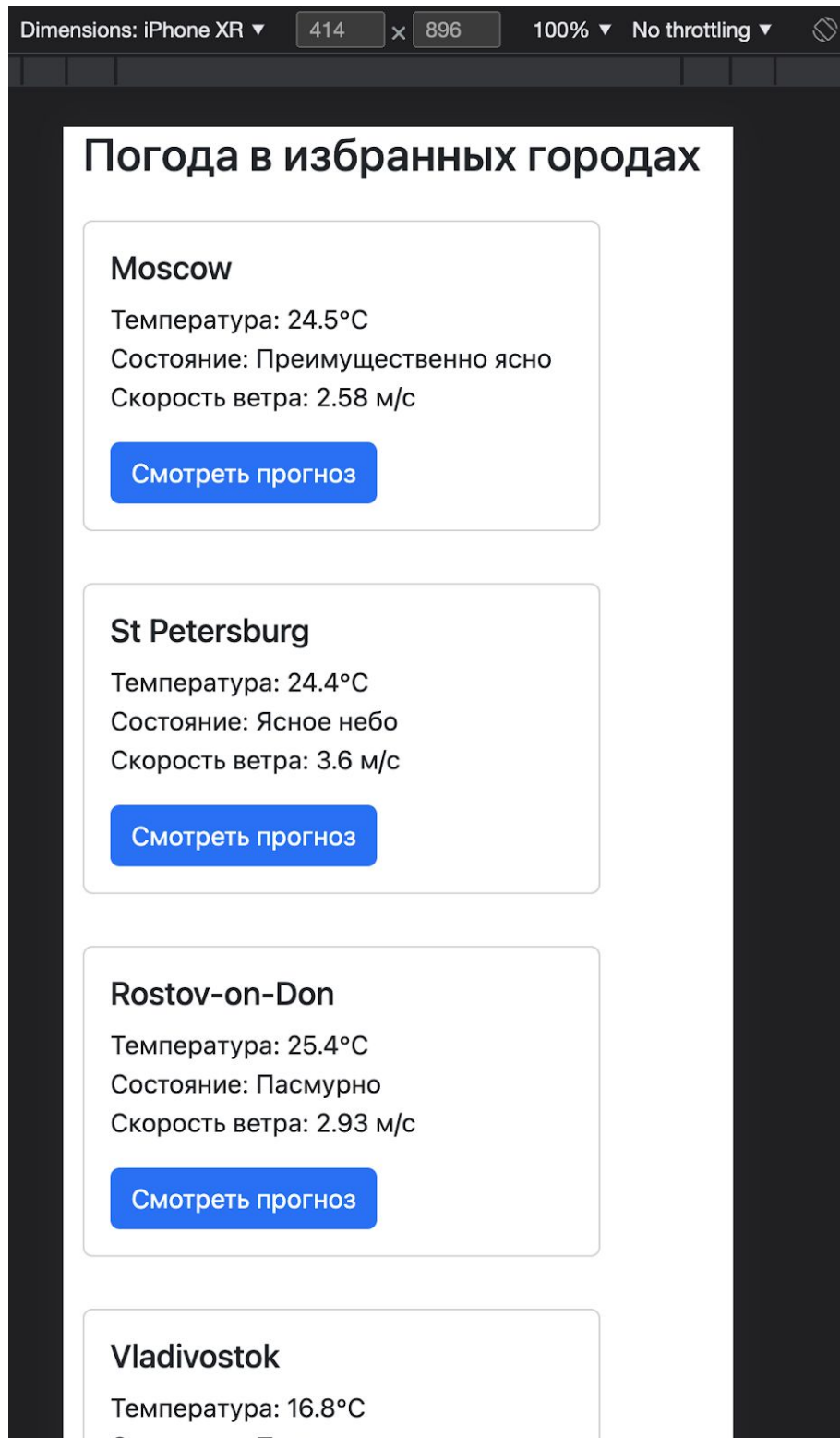
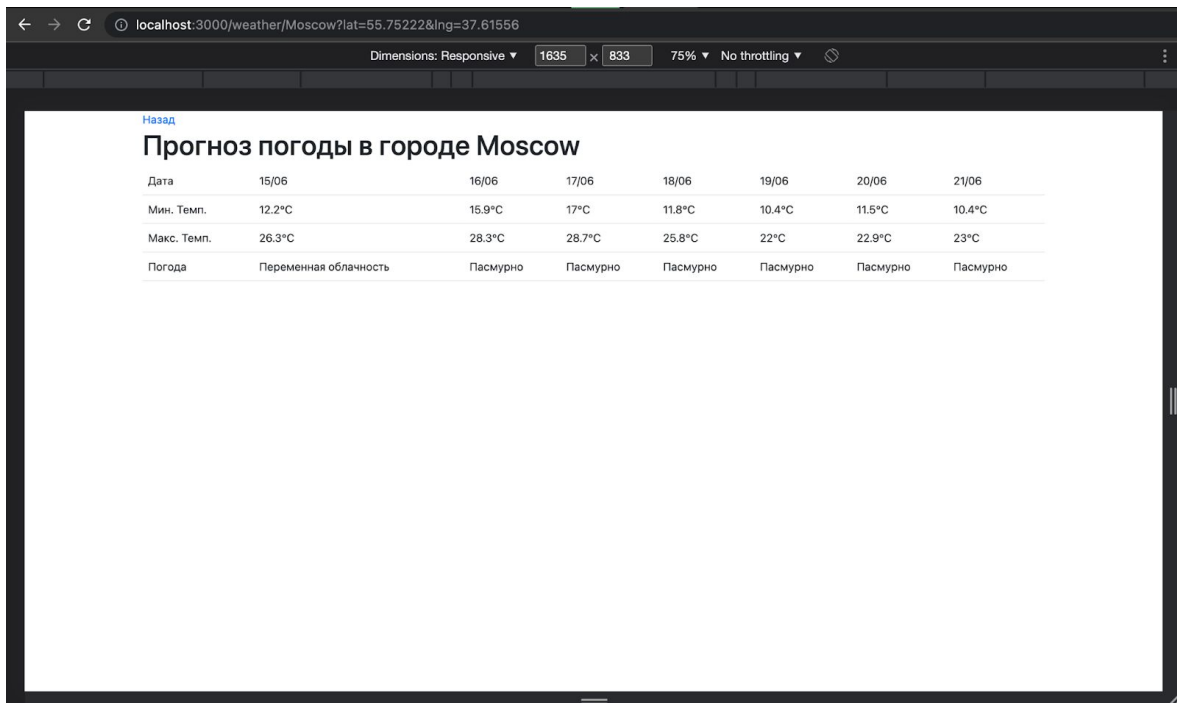


Рисунок 3. **Десктопный вид страницы прогноза погоды.**

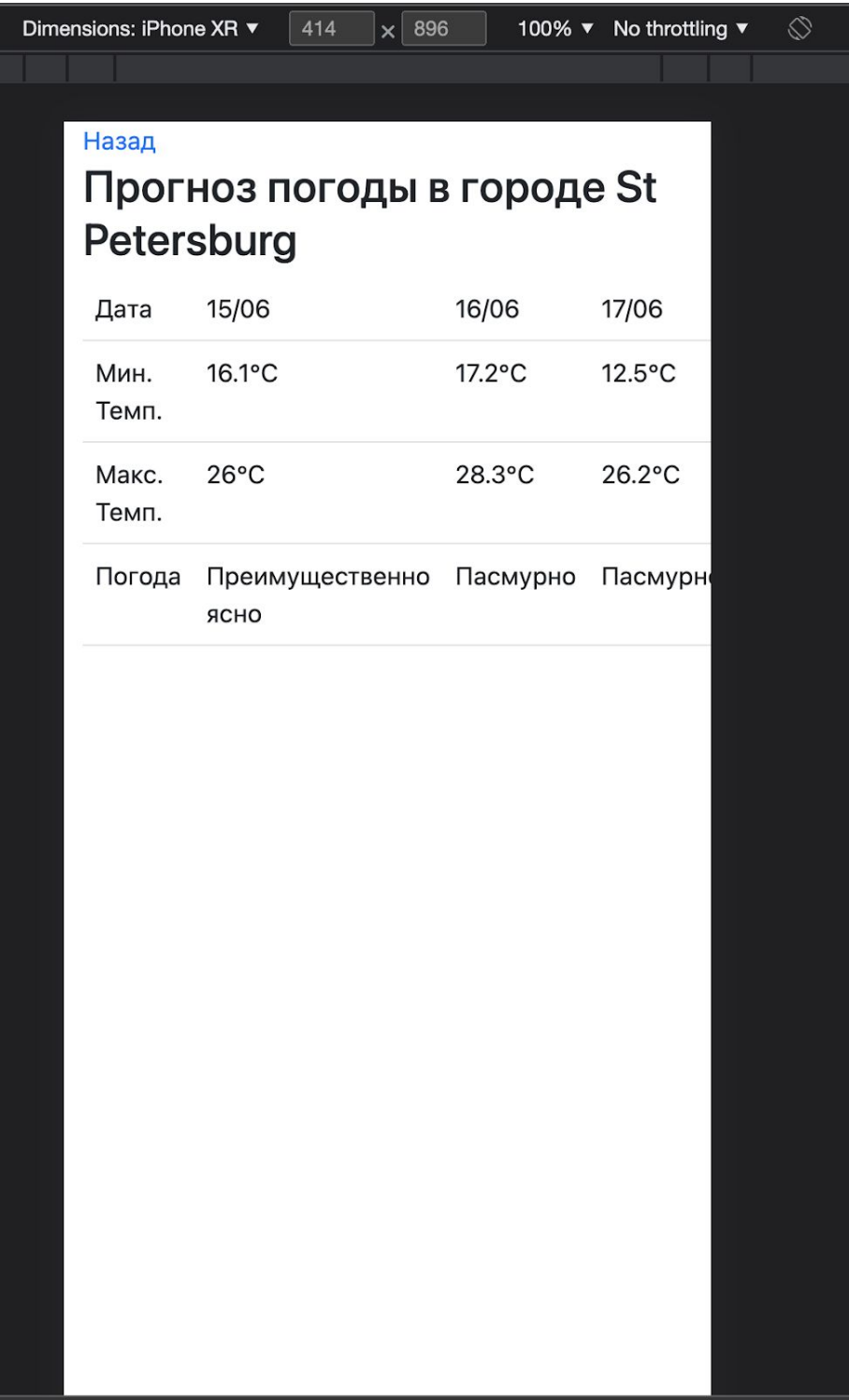


Назад

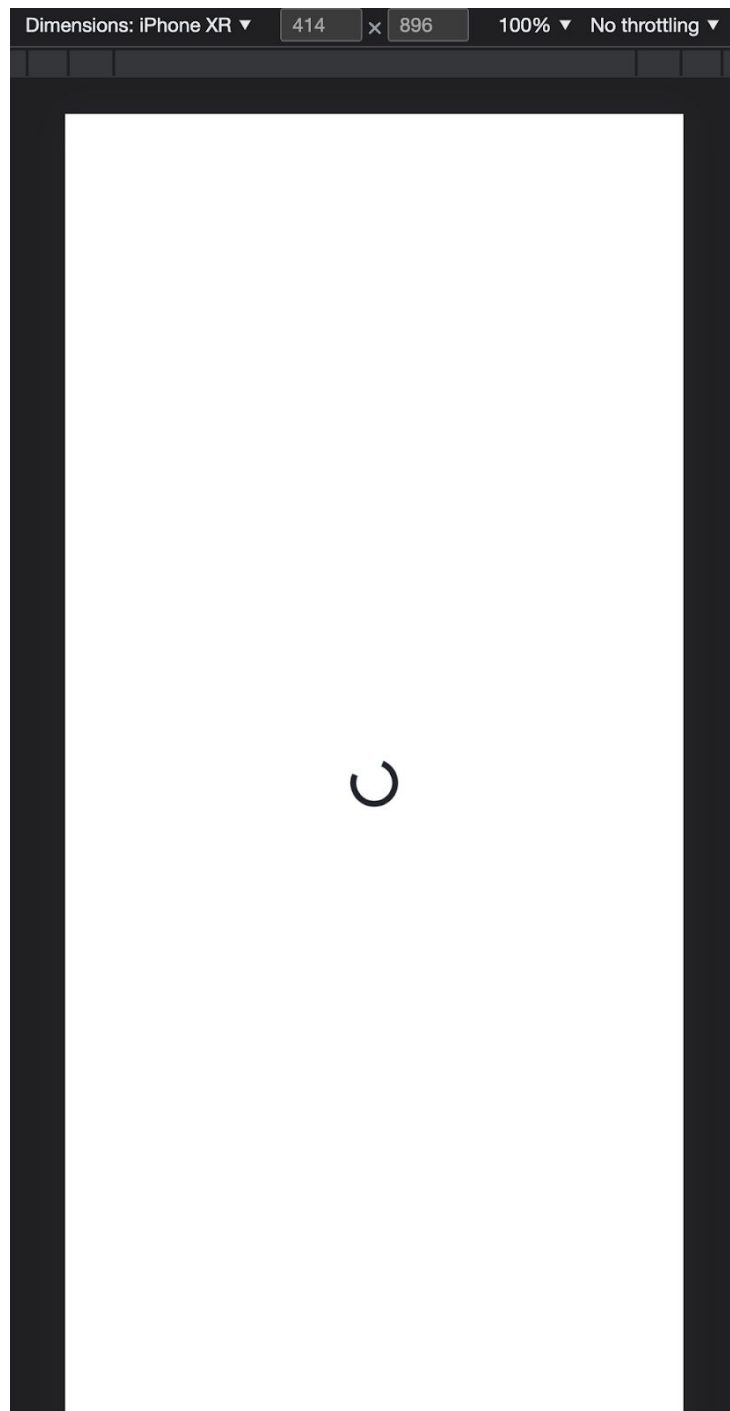
### Прогноз погоды в городе Moscow

Дата	15/06	16/06	17/06	18/06	19/06	20/06	21/06
Мин. Темп.	12.2°C	15.9°C	17°C	11.8°C	10.4°C	11.5°C	10.4°C
Макс. Темп.	26.3°C	28.3°C	28.7°C	25.8°C	22°C	22.9°C	23°C
Погода	Переменная облачность	Пасмурно	Пасмурно	Пасмурно	Пасмурно	Пасмурно	Пасмурно

Рисунок 4. Мобильный вид страницы прогноза погоды. Таблица горизонтально скроллится



## Рисунок 5. Прелоадер



## Рисунок 6. **AutocompleteInput** на главной странице в десктопном виде

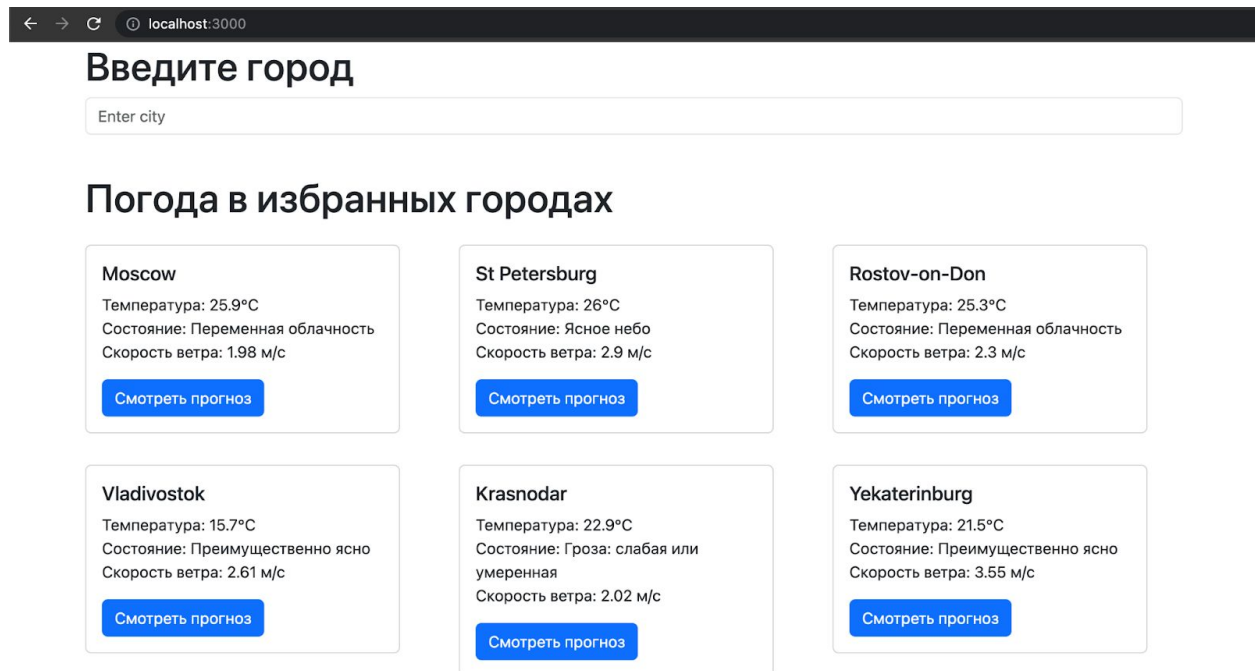


Рисунок 7. **AutocompleteInput** с выпадающим списком вариантов в мобильном виде

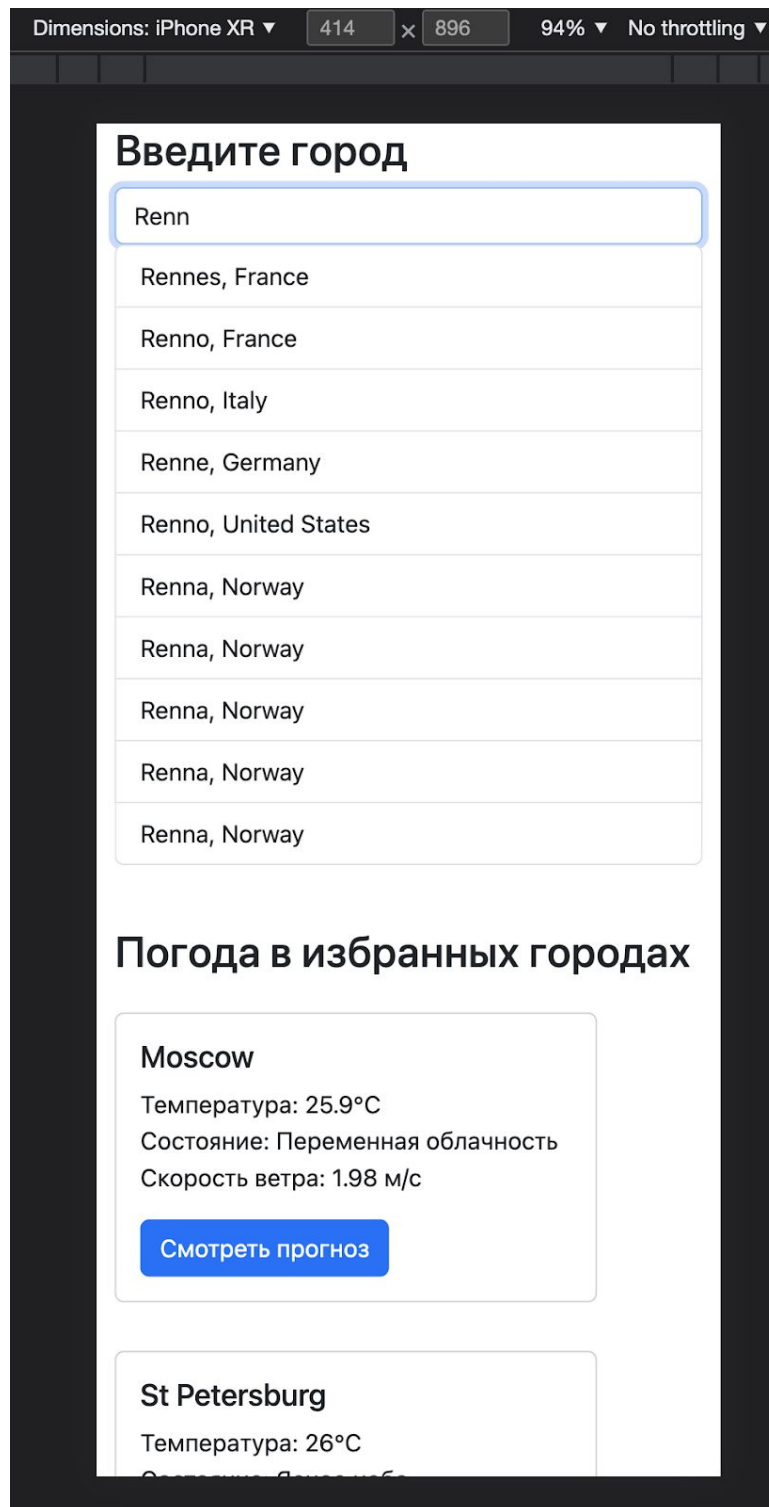


Рисунок 8. Переход на страницу прогноза погоды по клику на выпадающий вариант в AutocompleteInput

