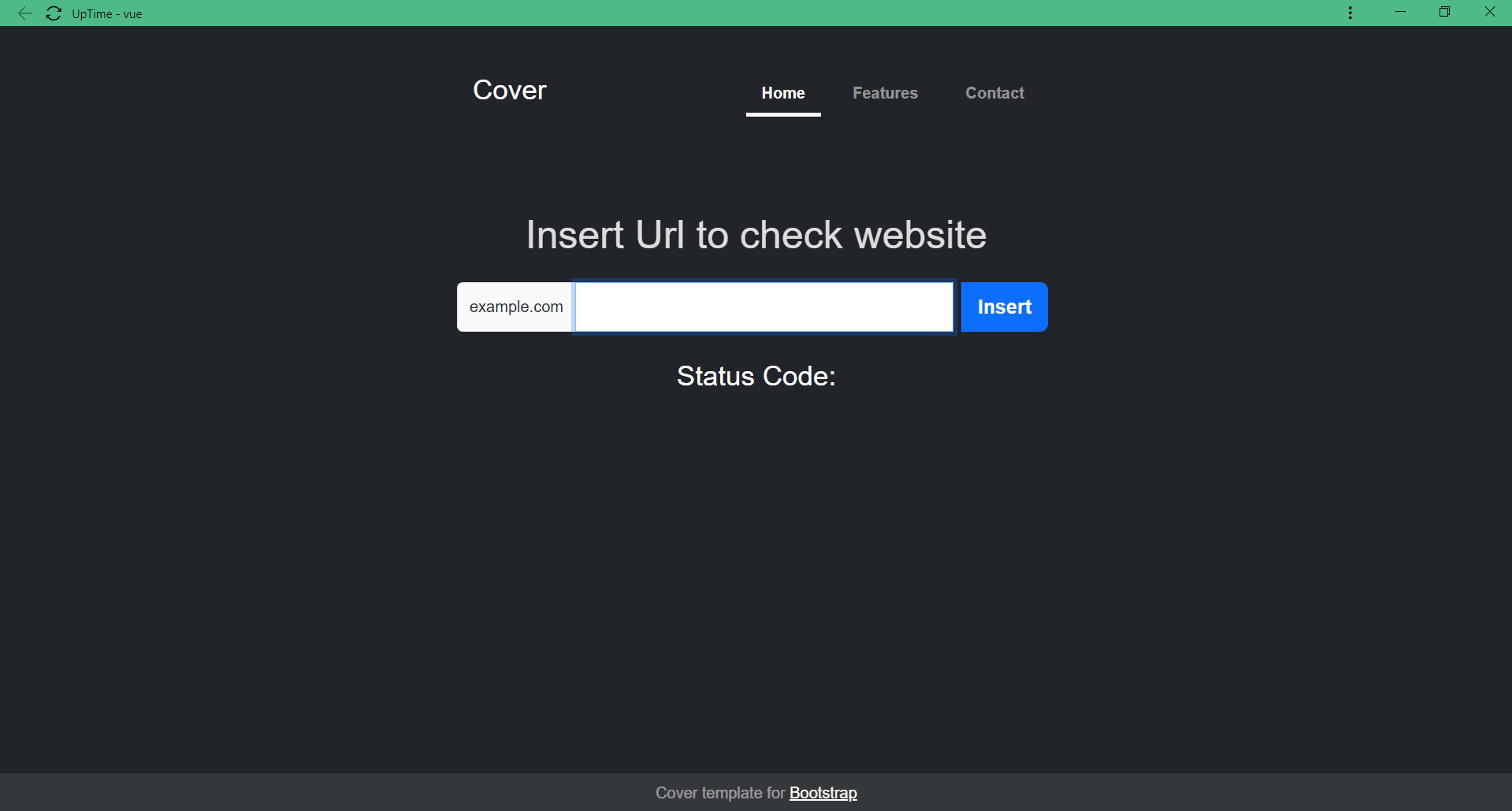
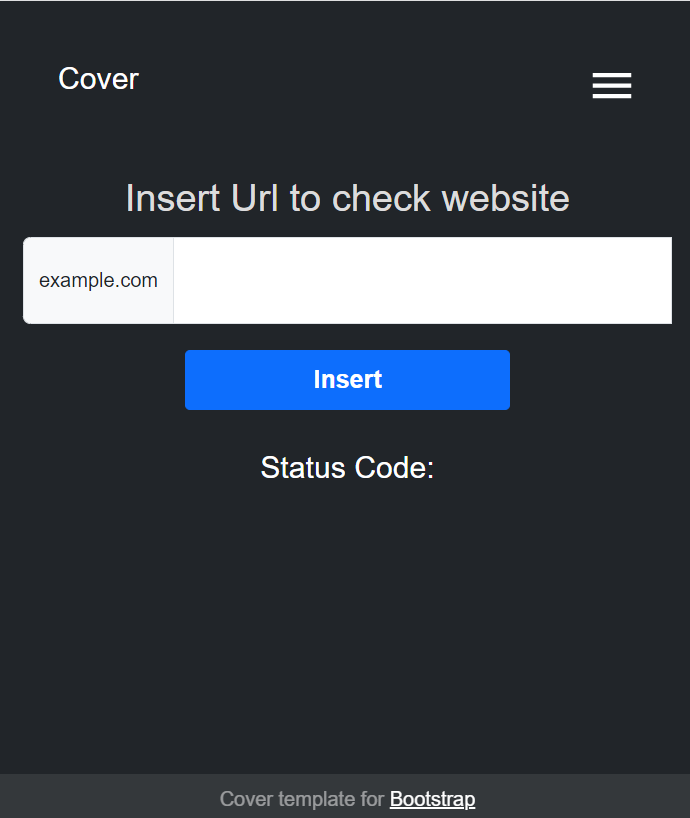
**Uptime**

Aпликација за проверка на availability/uptime на сервиси мобилна и веб



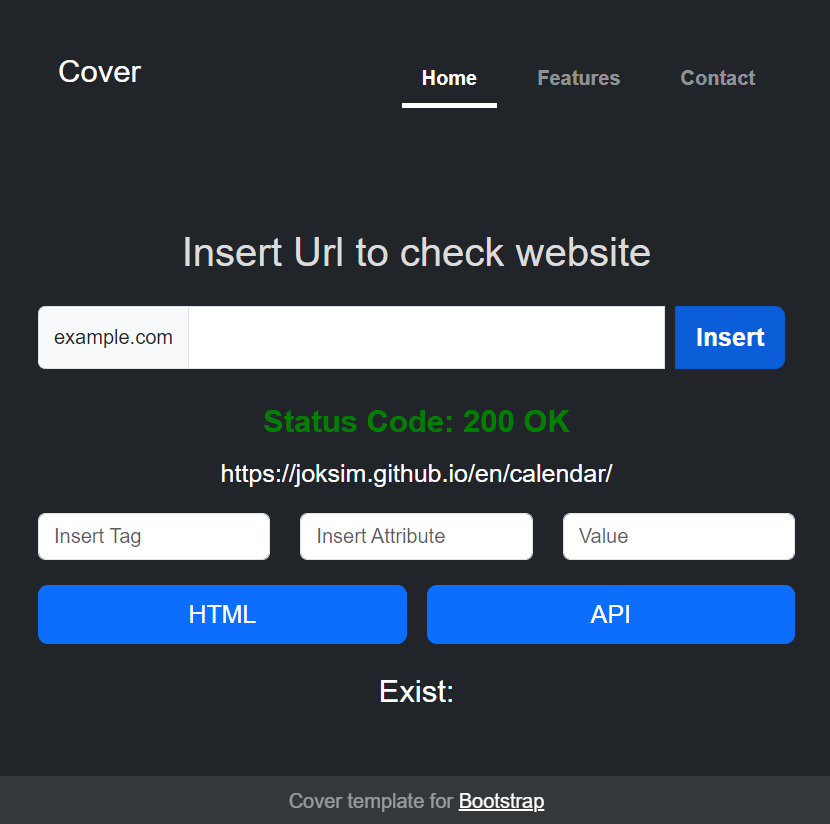
Првичен изглед на интерфејсот за PC



Првичен изглед на интерфејсот за мобилен телефон

**Чекори кои треба да ги следиме за правилно да ја користеме апликацијата:**

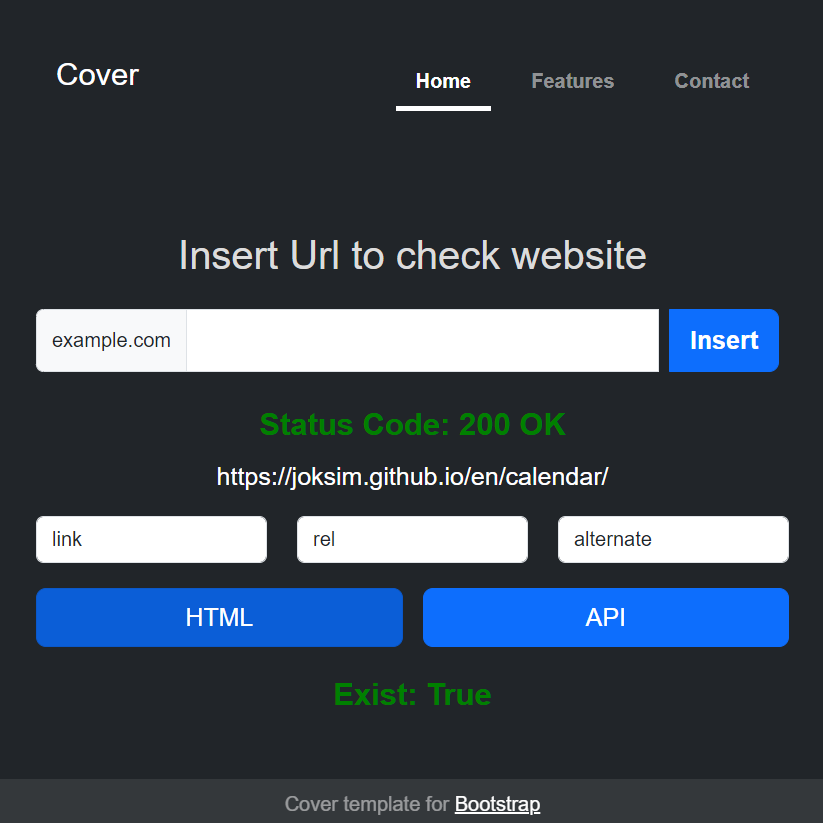
* **Чекор 1: Внесуваме валидна URL адреса на input полето и притискаме на копчето “Insert”.**
  + За пример ќе ја земеме URL адресата: <https://joksim.github.io/en/calendar/>



Во овој момент дознаваме дека Веб страницата е валидна бидејќи имаме

**Status Code: 200 OK**

* **Доколку веб страницата е валидна, автоматски ни се овозможуваат нови полиња кои треба да ги пополнеме за понатамошна проверка.**
* **Чекор 2: Во овој чекор треба да одлучиме дали сакаме да провериме вредности од *API* или од *HTML* страница.** 
  + **Во нашиот случај избираме HTML. Во останатите input полиња ќе внесеме податоци за Tag: link, Attribute: rel, Value: alternate**



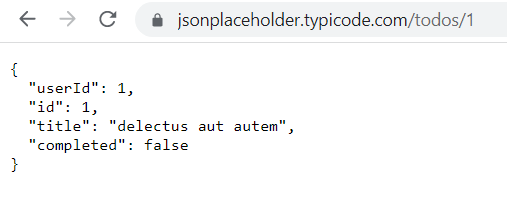
Во овој момент дознаваме дека постои

<link rel="alternate">

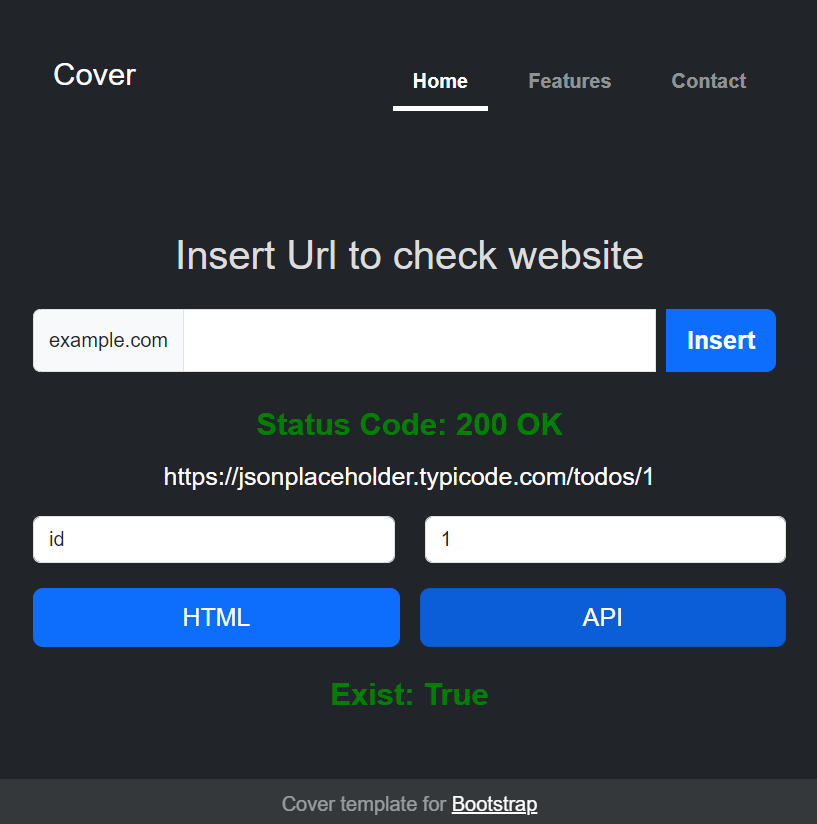
Во HTML страницата бидејќи

**Exist: True**

* **Чекор 3: Проверка на API - клуч: вредност**
* За пример ќе го земеме ова API: https://jsonplaceholder.typicode.com/todos/1
* Имаме JSON формат:



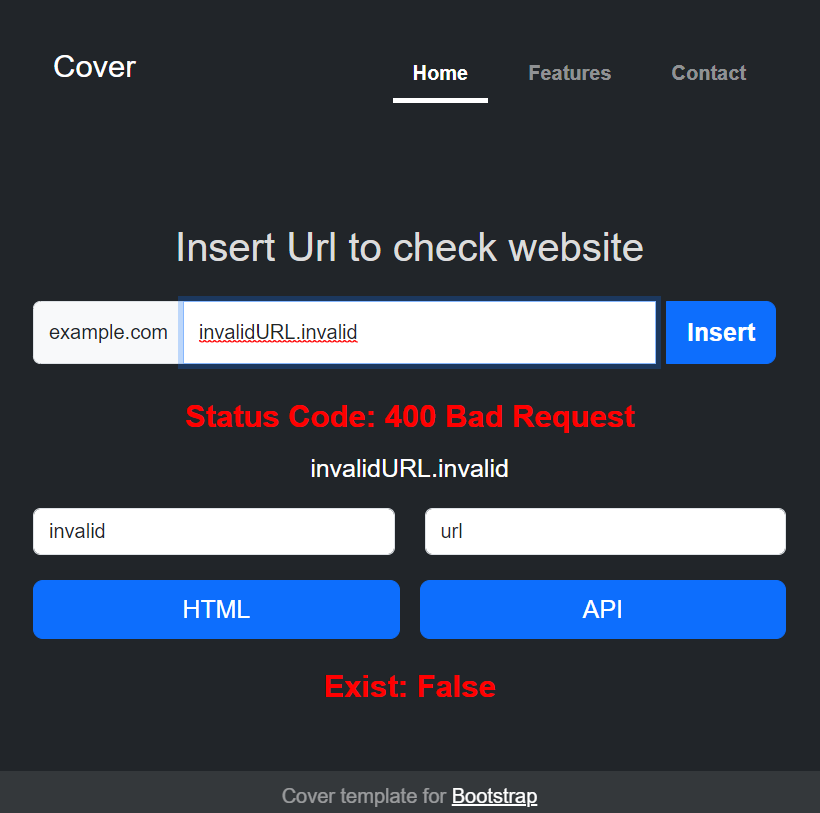
* Следно ќе провереме дали постои
  + - *Key*: “id”, *Value*: “1”



Добивме Status Code: **200 OK** и **Exist: True**

Тоа значи дека имаме валидно **API** и постои клучот **“id”** со вредност **“1”**

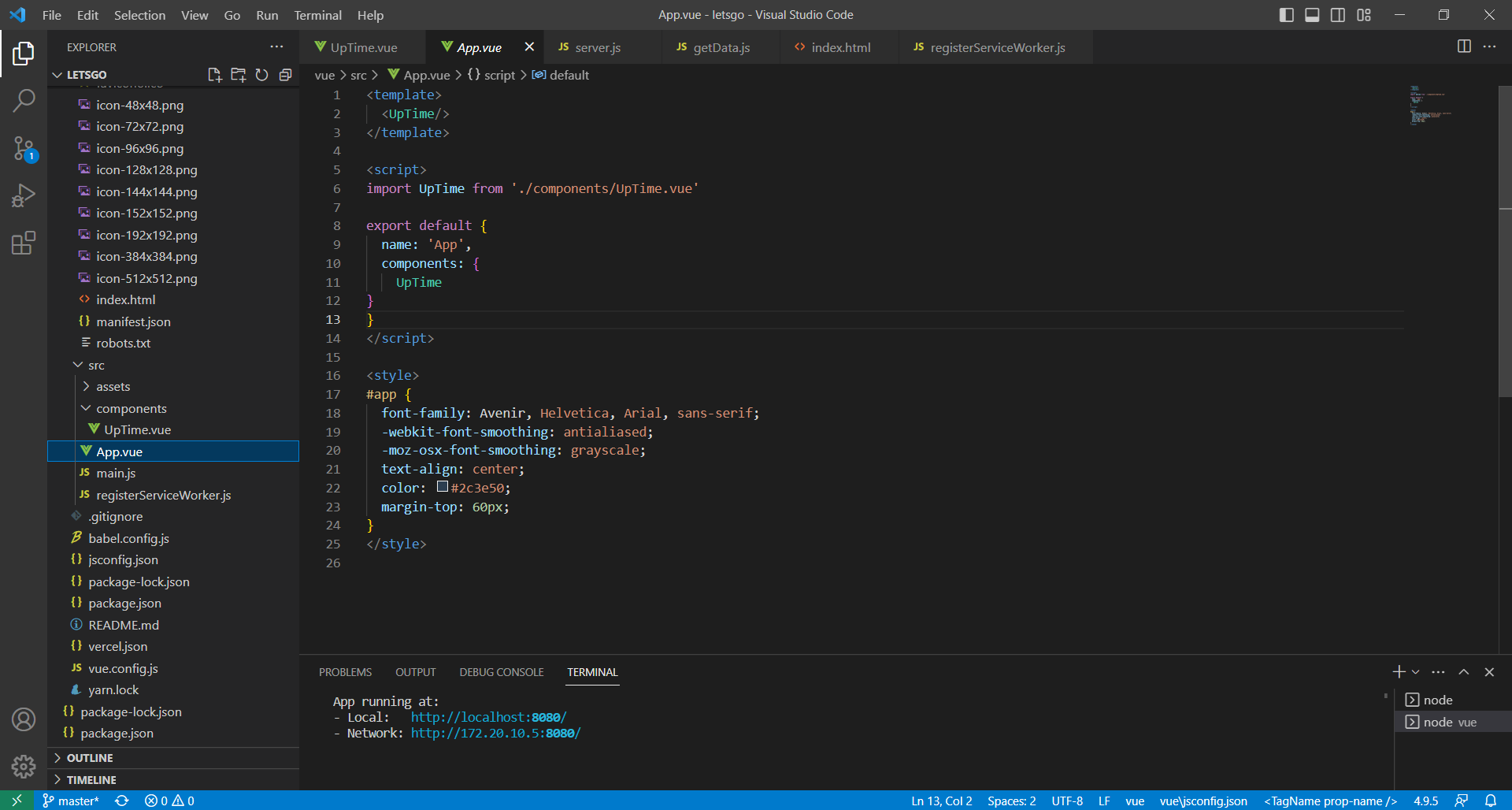
Во најлошо сценарио можеме да добиеме **Status Code: 400 Bad Request** и **Exist: False** што значи дека имаме погрешно внесено URL адреса или имаме непостоечни вредности ставено во input полињата.



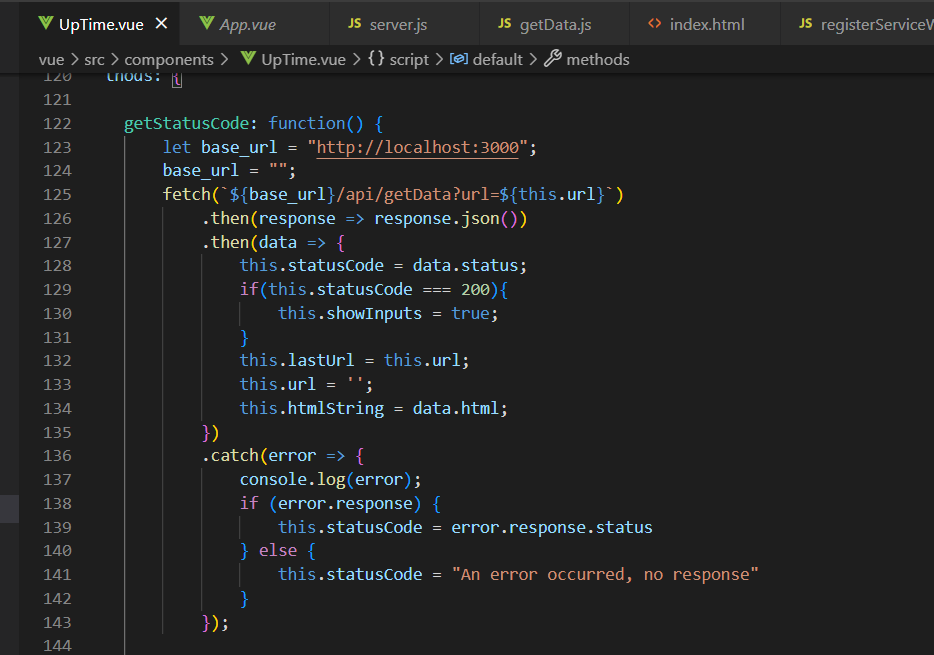
Вака изгледа кога имаме погрешни податоци внесено во апликацијата

**CODE:**

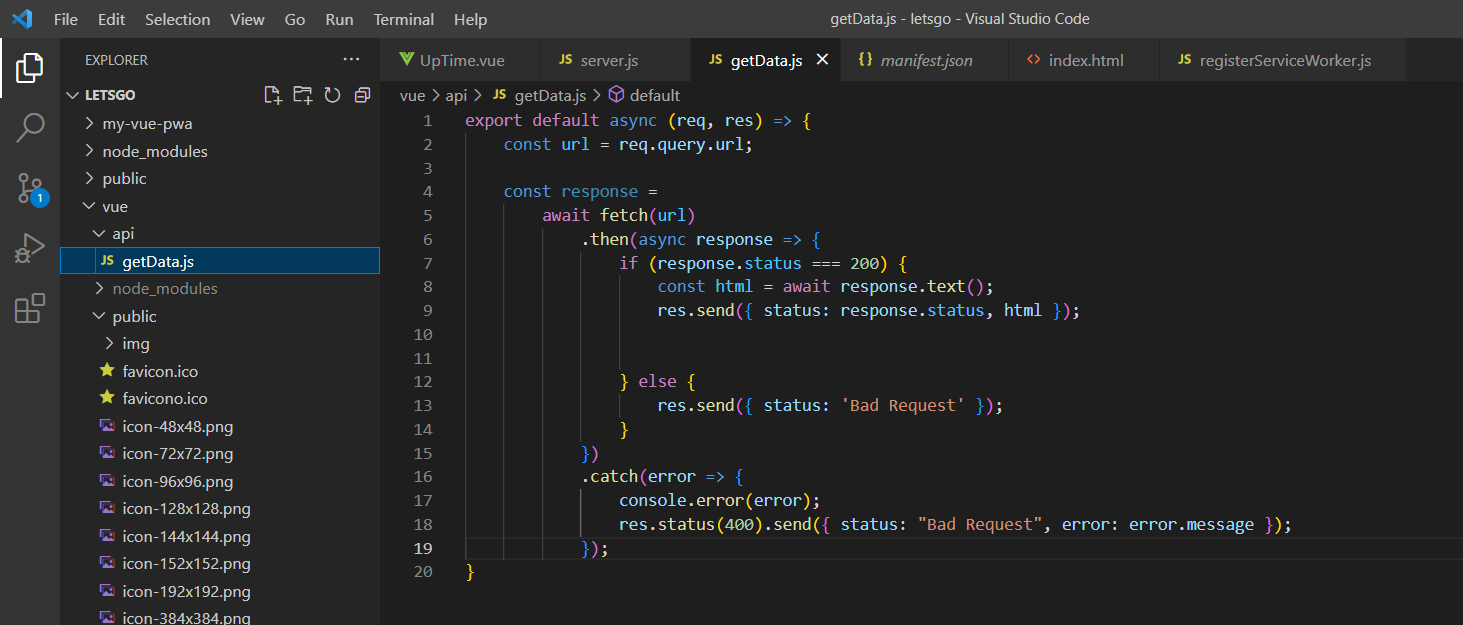
* **Frontend на апликацијата: Vue.js**
* **Користиме една компонента која е именувана како UpTime.vue**



* **Во UpTime.vue кога корисникот ќе внесе URL адреса и притисне на копчето “Insert”, се повикува функцијата getStatusCode()**

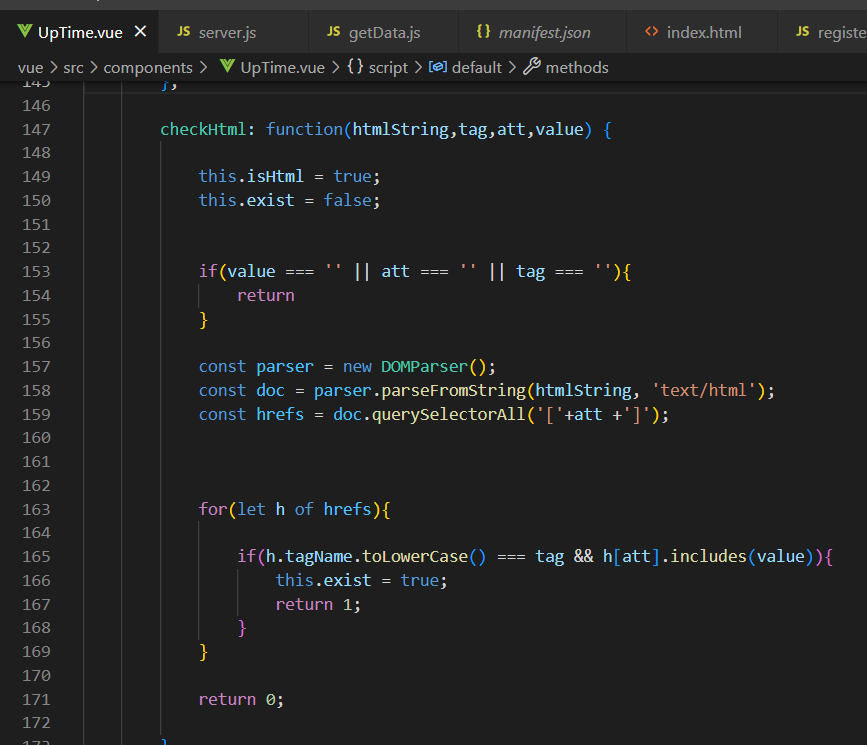


* **Со функцијата GetStatusCode() праќаме Get Request до /api/getData со тоа што ја праќаме URL адресата од input полето. Потоа ги зема податоците кои ни се вратени од /api/getData и правиме проверка дали статус кодот е 200, доколку не е статус кодот 200 праќаме error message, во спротивно праѓаме статус код: 200 и ги овозможуваме останатите input полиња.**
* **Зошто ни е потребно да праќаме URL адреса преку /api/getData?**
  + Ако пратеме get request преку фронтенд, т.е преку **UpTime.vue**, нè блокираат Веб страниците со **CORS**, а доколку пратеме get request од getData.js која е извршена како **ламбда функција** од **Vercel** успешно се праќа get барањето. Со тоа успешно го добиваме статус кодот од get барањето од Веб страницата.



* + **Во /api/getData ја зимаме URL адресата од request-от и праќаме get барање то таа URL адреса. Доколку добиеме статус код 200, тогаш го праќаме назад кон UpTime.vue како response со што го праќаме и целиот HTML/API документ. Доколку не добиеме статус код 200, враќаме назад како response status code 400 и error message.**
  + **Во UpTime.vue компонентата имаме уште 2 функции: checkHtml() и checkApi()**

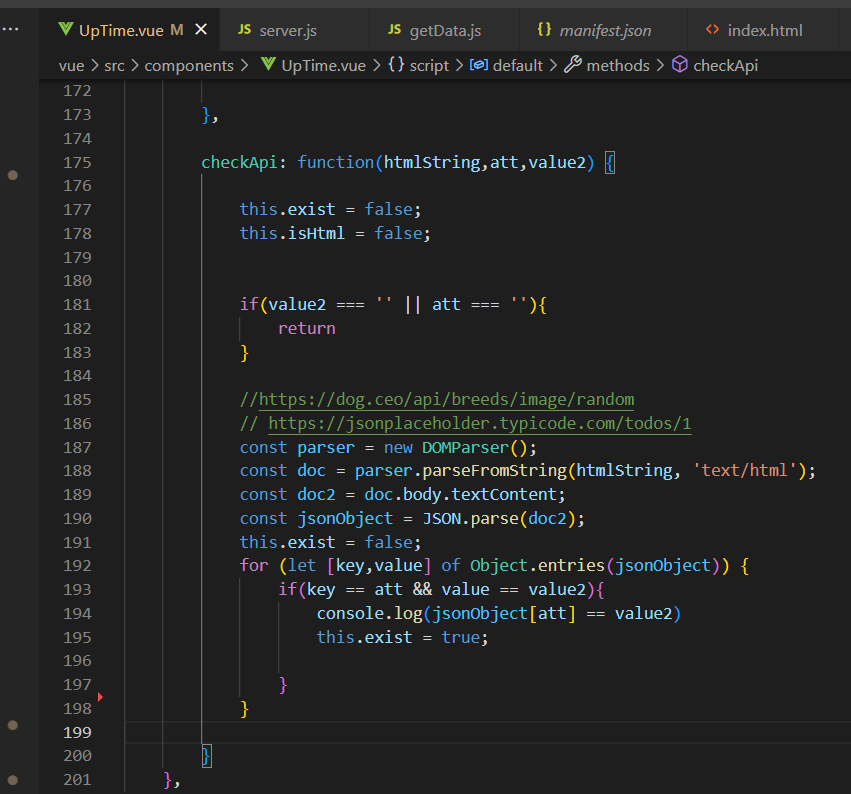
**checkHtml()**



* **Оваа функција прима 4 аргументи:**

**htmlString, tag, attribute, value. Од дадениот HTML документ ги листа сите tags и проверува дали на одреден tag постои соодветен attribute со соодветна value.**

**checkApi()**



* **Оваа функција прима 3 аргументи:**

**htmlString, key, value. Од даденото API ги листа сите вредности и проверува дали за одреден key постои соодветна value.**