Small Cafe

Introduction

System for a small café will be presented here. It uses Client-server architecture along with a database.

Database

Database is realized via MySQL. It has 4 tables with the following purposes:

- **article** – List of articles and prices per article. Articles can be temporarily deactivated.
- **employee** – email and password for logging employed waiters.
- **guestorder** – Order made by guest. It has unique id, table, time and final price. Via unique id it is connected with one or more **articleinorder**.
- **articleinorder** – His parent table is one **guestorder**. Contains articles in guest order. It contains detail of an article in list of articles which guest ordered.

A script *smallcafedb.sql* for creating a database is included within project files.

Server

Server-side application is based on Node.js. It is node project, but in essence it is Representational State Transfer (REST) API via HTTP. It allows us usage of GraphQL, which is a query language for APIs and a runtime for executing queries on data which is stored in MySQL database. On server there are following features:

- express – used for creating and hosting web application on node.
- graphql – used for testing purposes (see picture 1.)
- mysql – node.js driver for mysql
- cors –package for providing a Connect/Express middleware. Enables Cross-Origin Resource Sharing (CORS).

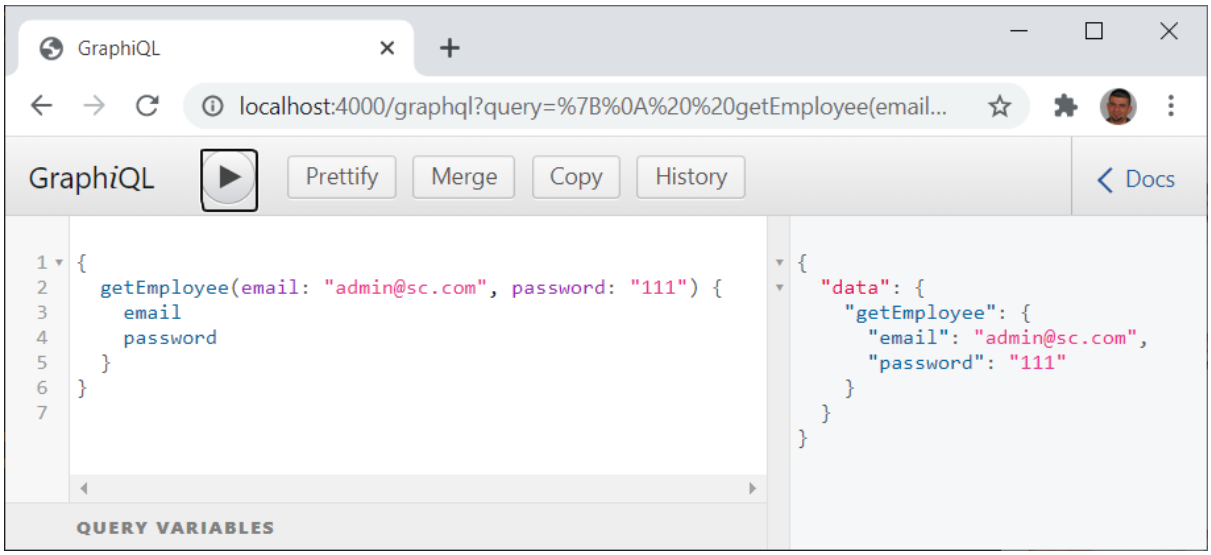


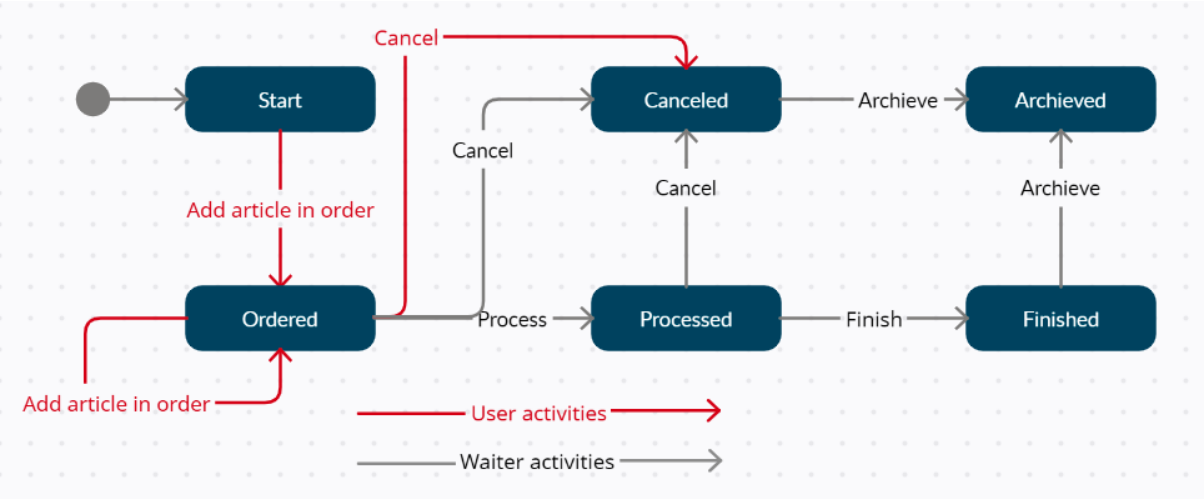
Image 1. GraphiQL

Client

Client is a React web application. It uses React Apollo to fetch data from GraphQL server (Node.js Server previously described). State management and other React features are realized via hooks. It also uses bootstrap capabilities. The popular front-end framework React Bootstrap is used. Also reactstrap is used for some components.

There are two roles: customer and waiter. Waiter must be authenticated using username and password. State diagram is shown in graph 1. Typical screens are shown in following images:

1. Image 2 - Start screen. Customer should select his table number.
2. Image 3 - Selecting articles. After selecting table customer will select articles in his order. He can make order or reset all.
3. Image 4 - Report about order details. User can cancel it also.
4. Image 5 - Waiter login on */waiter* path. Entry point for protected area.
5. Image 6 - Lists of orders in different states. States of order may be:
    a. Ordered. Orders made by customer, but not handled.
    b. Processed. Order accepted by waiter but still not delivered to customer.
    c. Finished. Order successfully delivered by waiter.
    d. Canceled. Order canceled by customer or by waiter.
    e. Achieved. Order is sent to achieve by waiter.
6. Image 7 - About page.
7. Image 8 - Contact page. Contains sample Google map.



Graph 1. State diagram (https://app.creately.com/diagram/NhEFnCF5czH/edit)

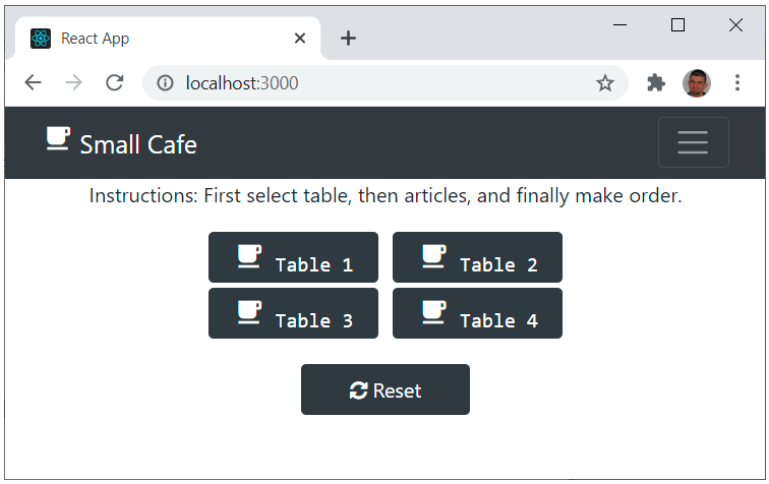


Image 2. Start screen

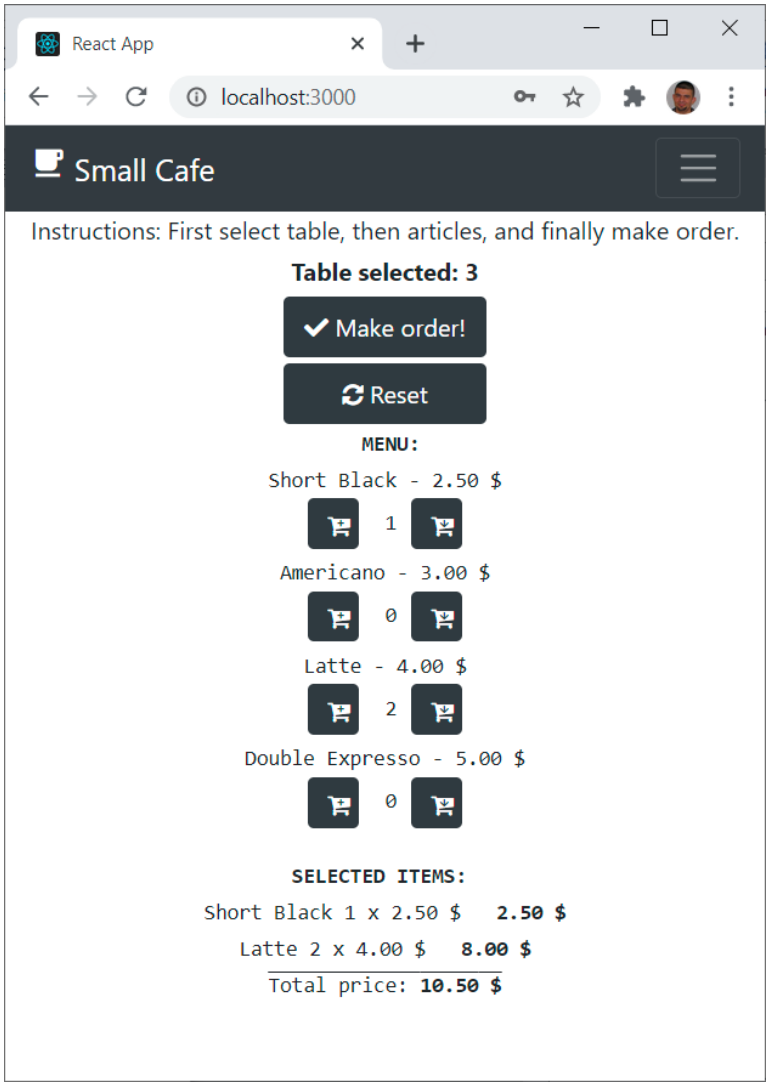


Image 3. Selecting articles

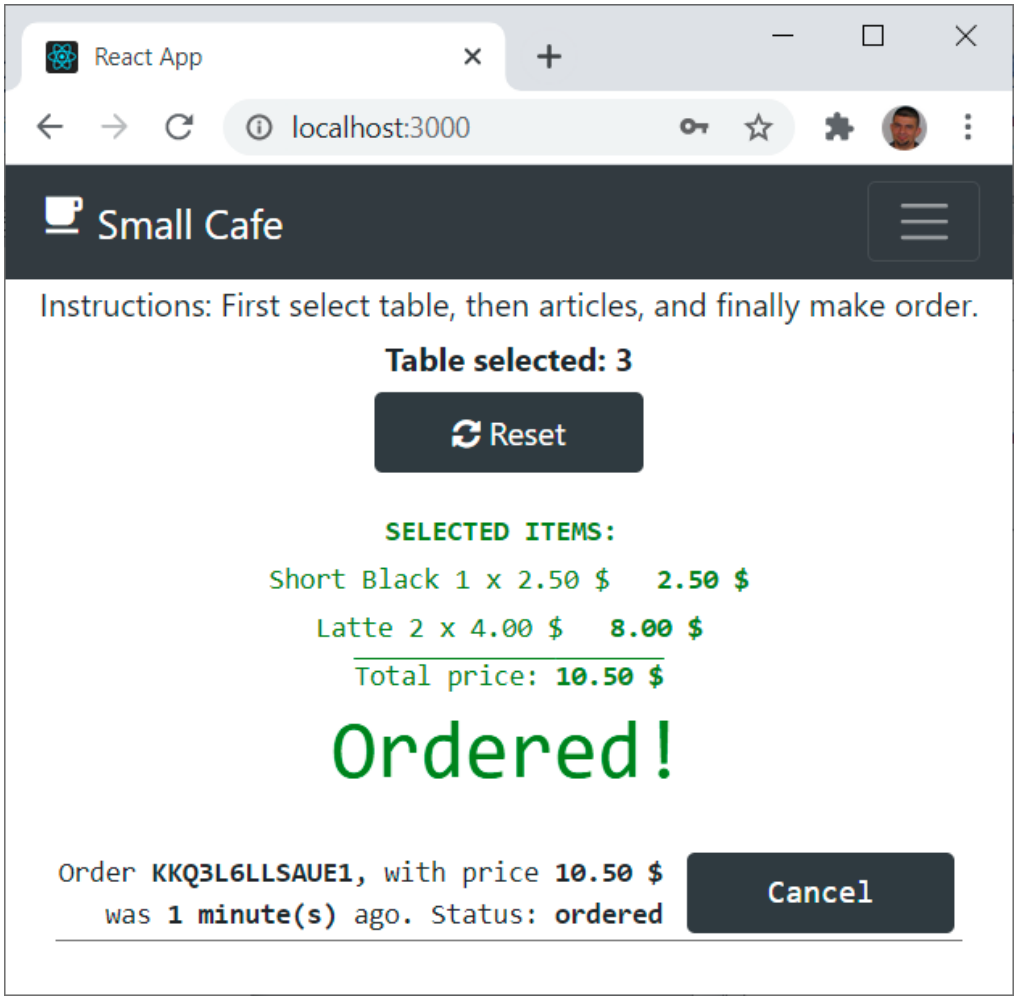


Image 4. Report about order.

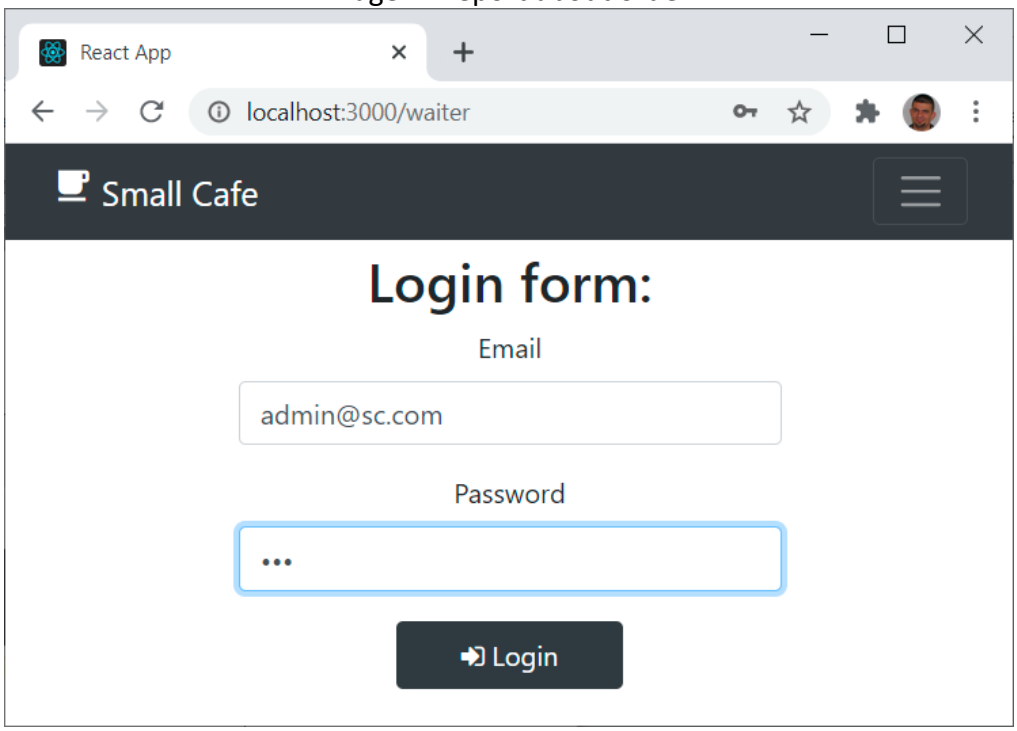


Image 5. Waiter login.

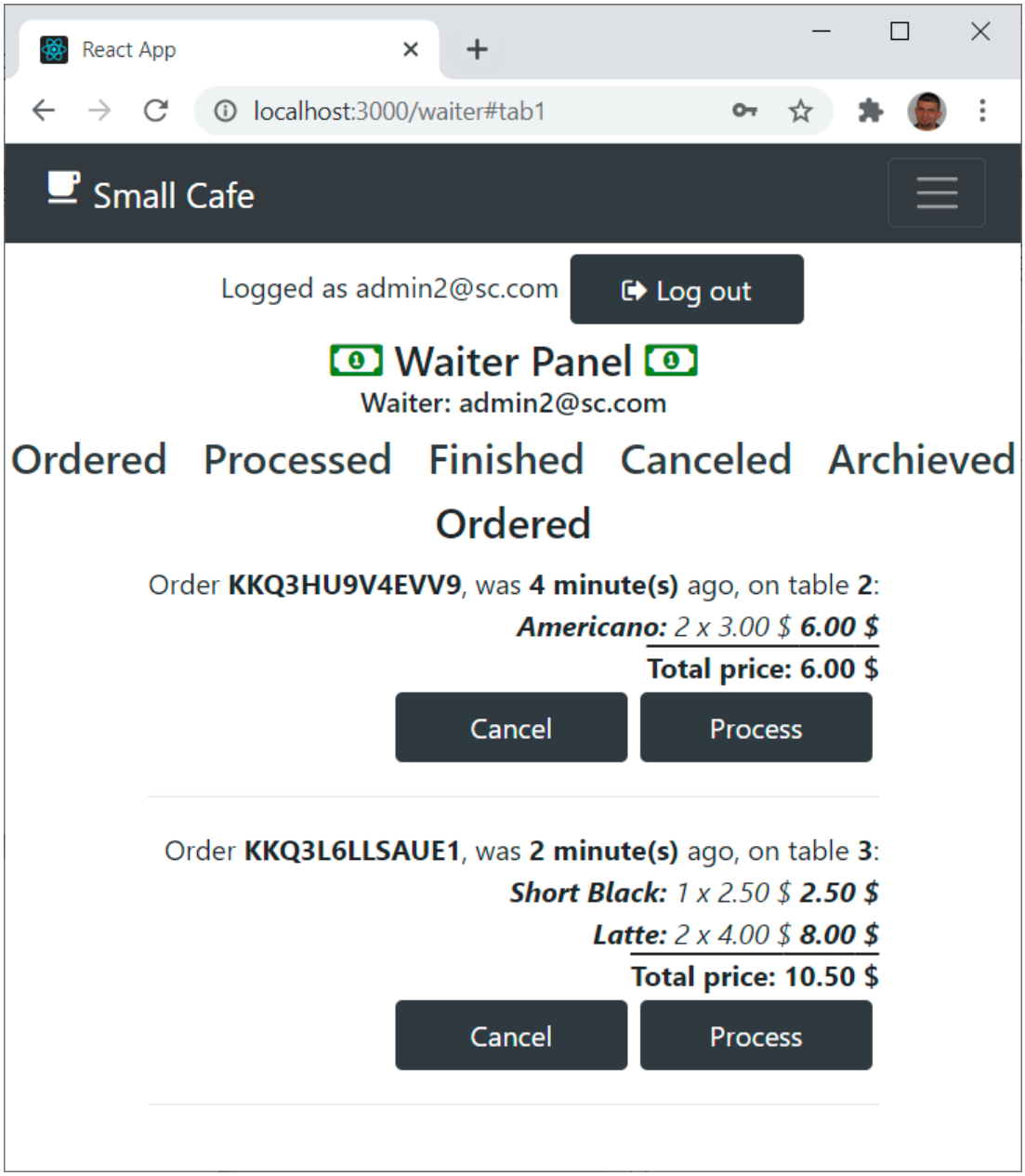


Image 6. Lists of orders in different states

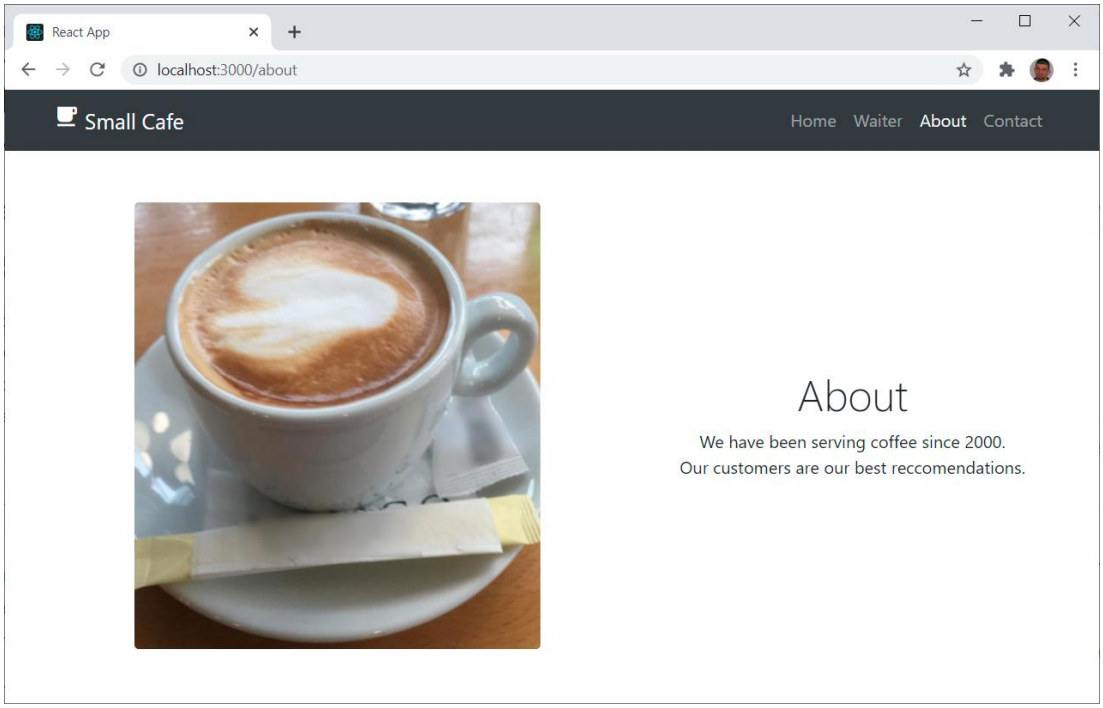


Image 7. About page

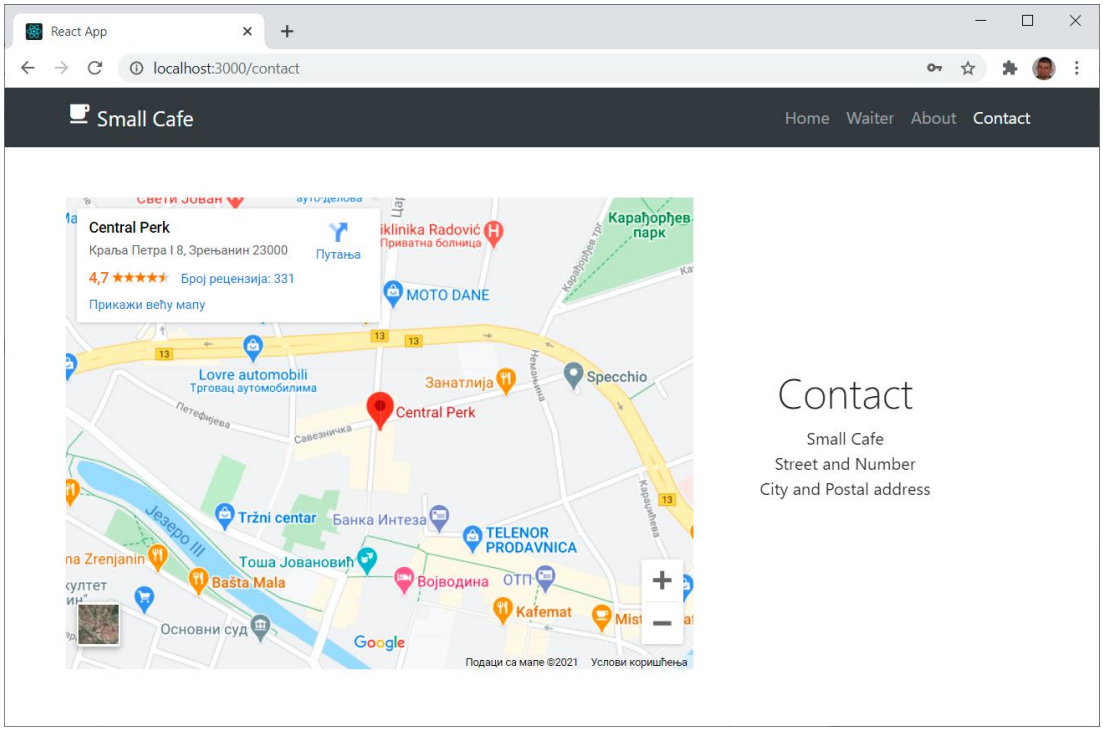


Image 8. Contact page