SDE INTERN ASSIGNMENT

Technical Requirements:

FRONTEND :

* ReactJs : Used for frontend framework
* Typescript : Used it because it helps in writing javascript code which is easier to debug and compile
* Material UI : Used to import buttons, tabs and other standard components
* React-Router : Used for routing through pages
* React-Redux : Used for state management of the application
* Redux-Thunk : Used as a middleware for redux
* Eslint@6.x : Used for running linting tests
* React-material-ui-form-validator : Used for running validation tests over form elements

BACKEND :

* NodeJs : Used for writing backend of the apps.
* ExpressJs : Used for writing the APIs
* Cors : Used to make cross origin requests between client and server
* Mysql : Package used for passing mysql queries over mysql database
* Nodemailer-sendgrid-transport : Used for sending emails on API call
* Nexmo : Used for sending SMS on API call

DATABASE:

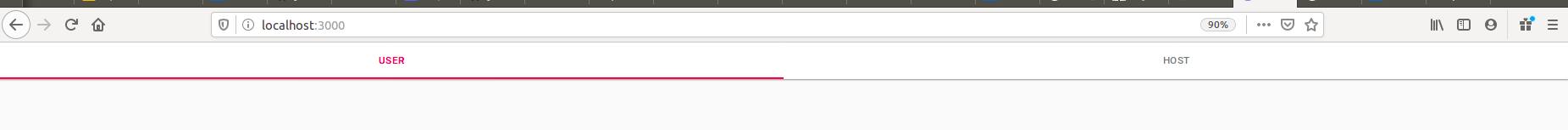
* MYSQL
* Mysql-workbench : Used to access the database on Ubuntu

Reason behind using the above stack :

I used ReactJs with typecript because it is easier to write code and helps in creating components with smooth transitions with much less lines of code. It is always better to use a backend framework that is built on the same language as it is easier to integrate it. So I used Express framework in the backend. I was equally convinced about using MongoDB or MYSQL here but because I already used MongoDB in my last project so I decided to use MYSQL so that I am well versed with its components as well.

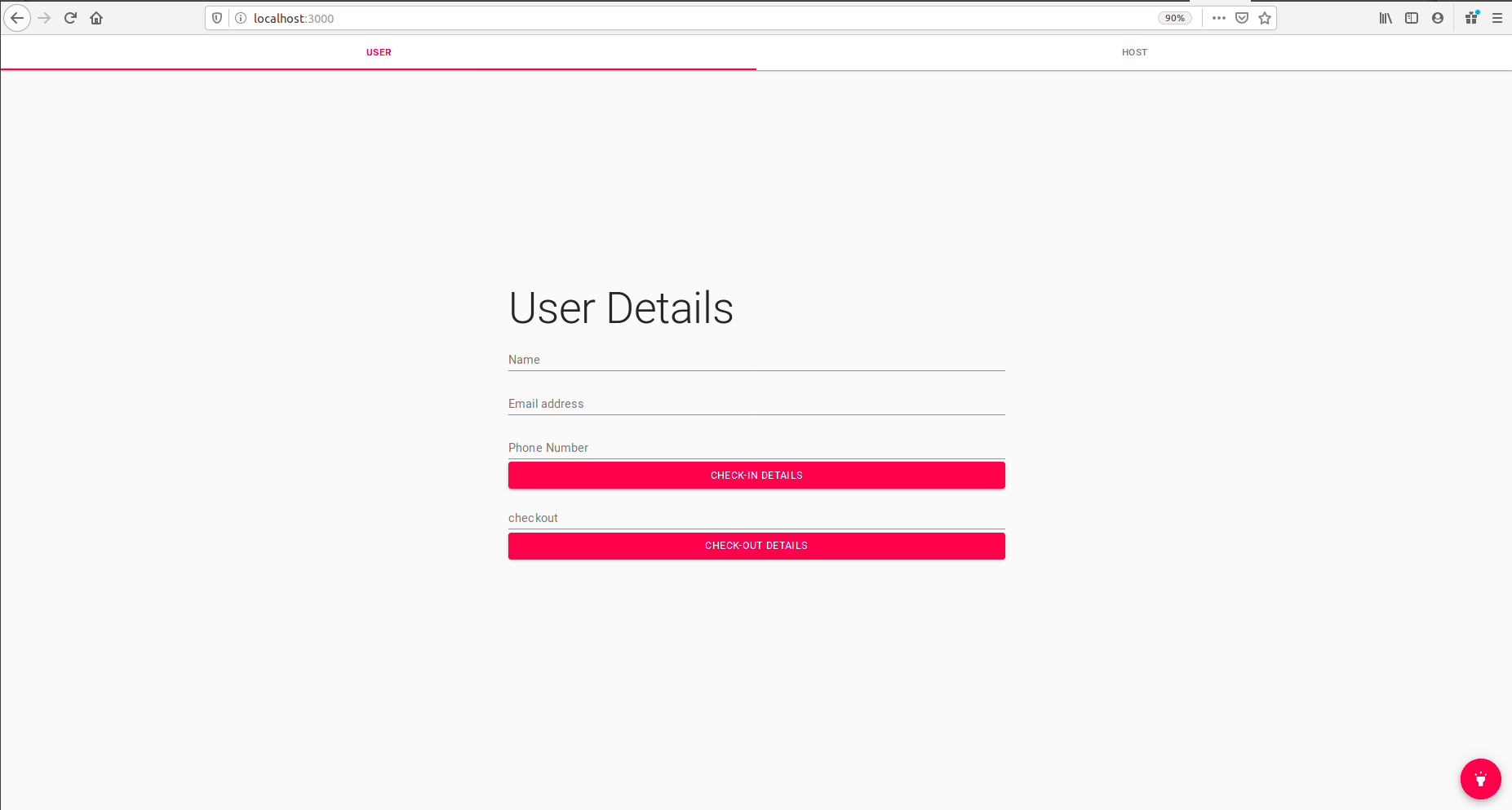
Approach:

* Created two tabs each for entering user details and host details one by one.

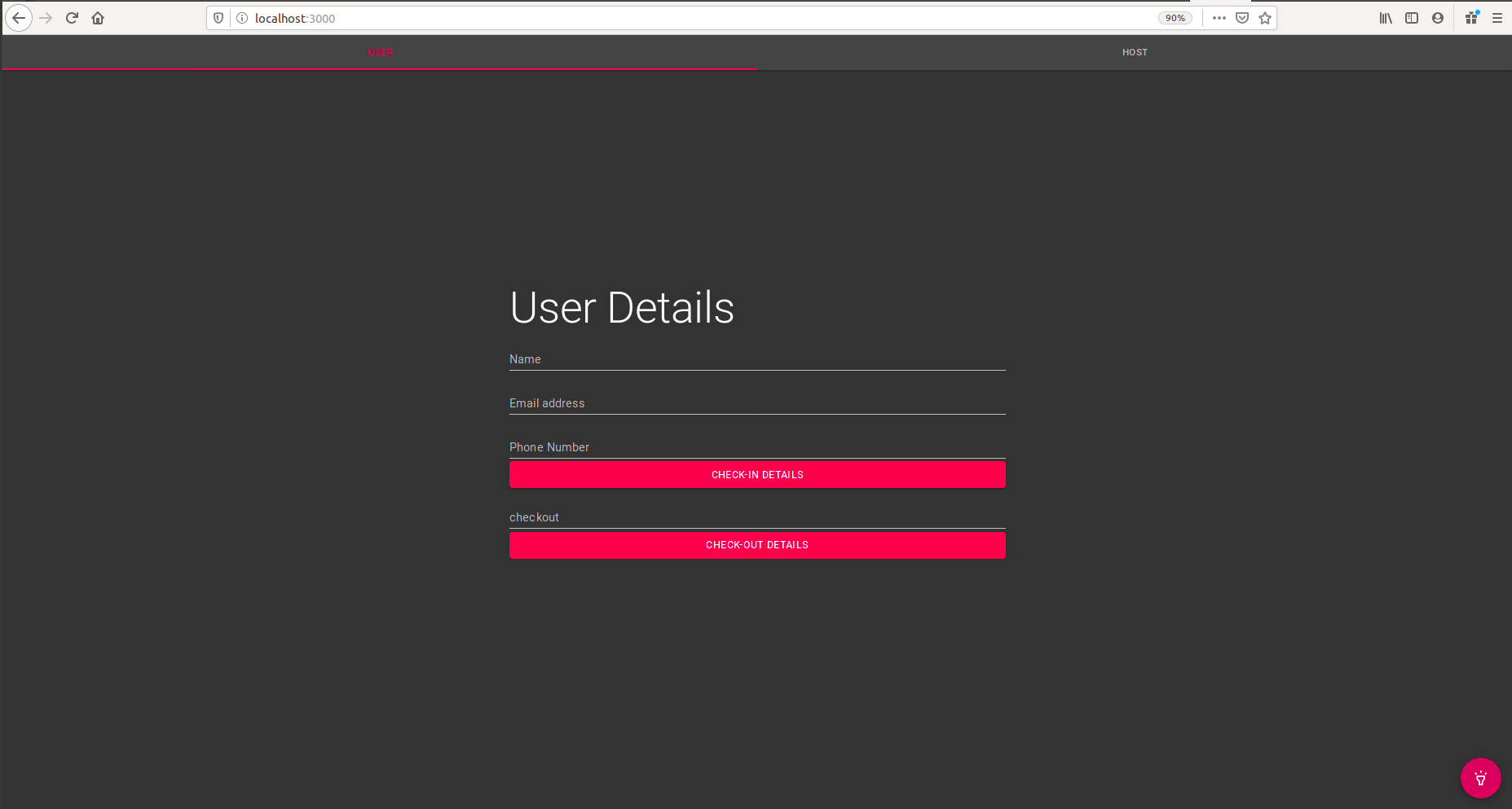


* Added a light-dark theme toggle button in the bottom right corner of the screen that provides the functionality to change the theme of the application.

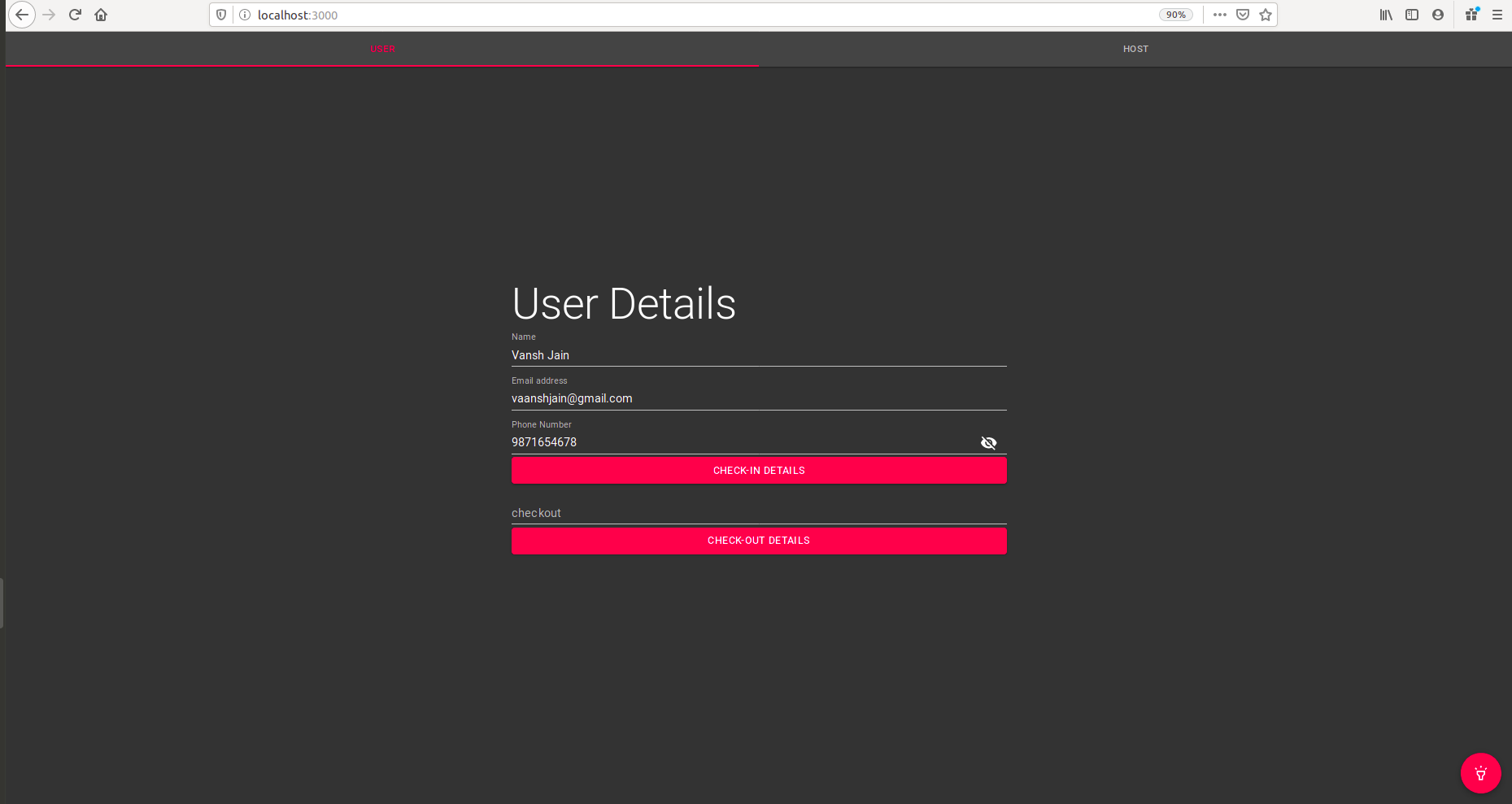
Light theme:



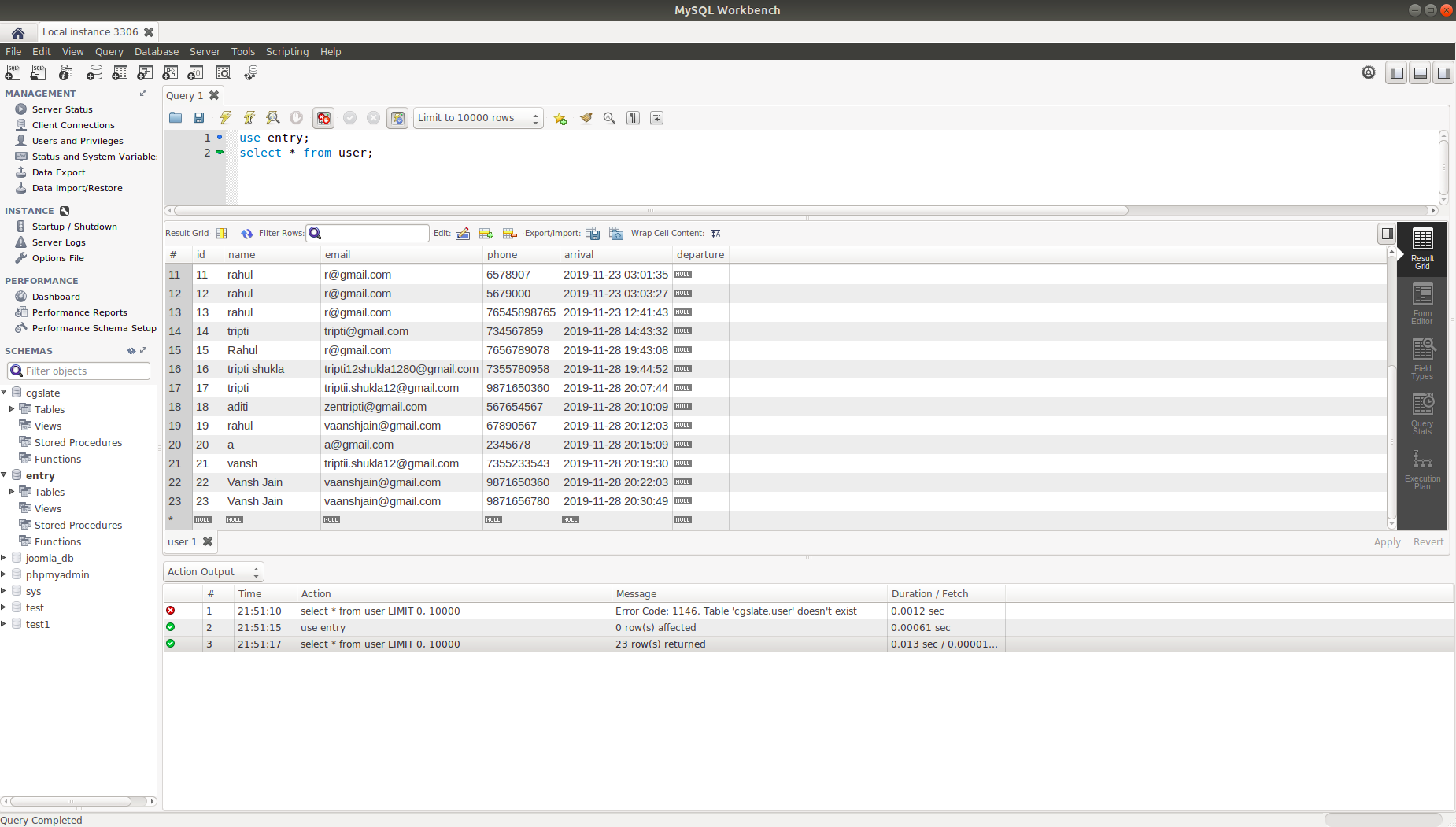
Dark Theme:



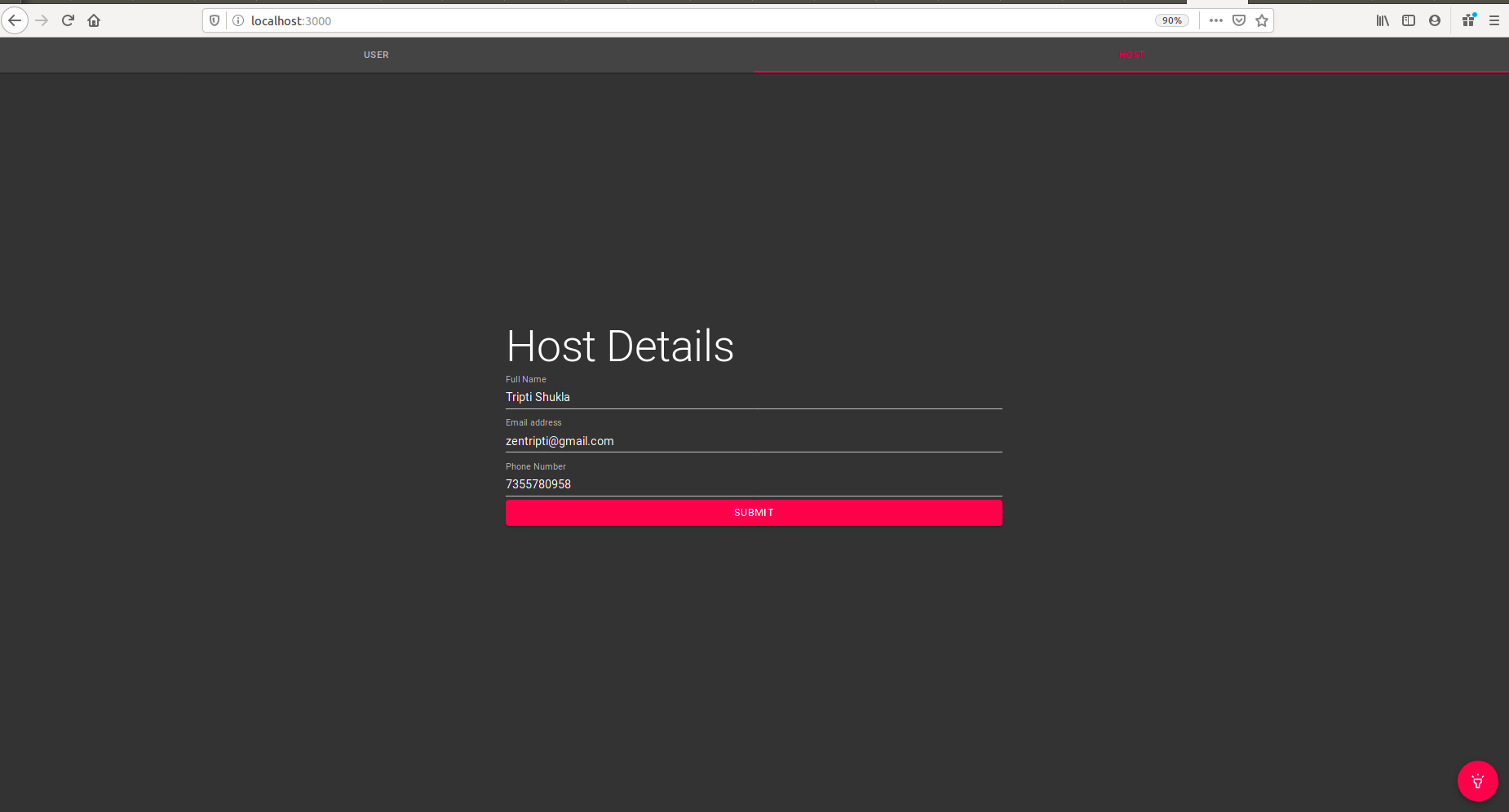
* Initially the user details tab was open and the visitor was asked to enter his/her details and press the submit button.



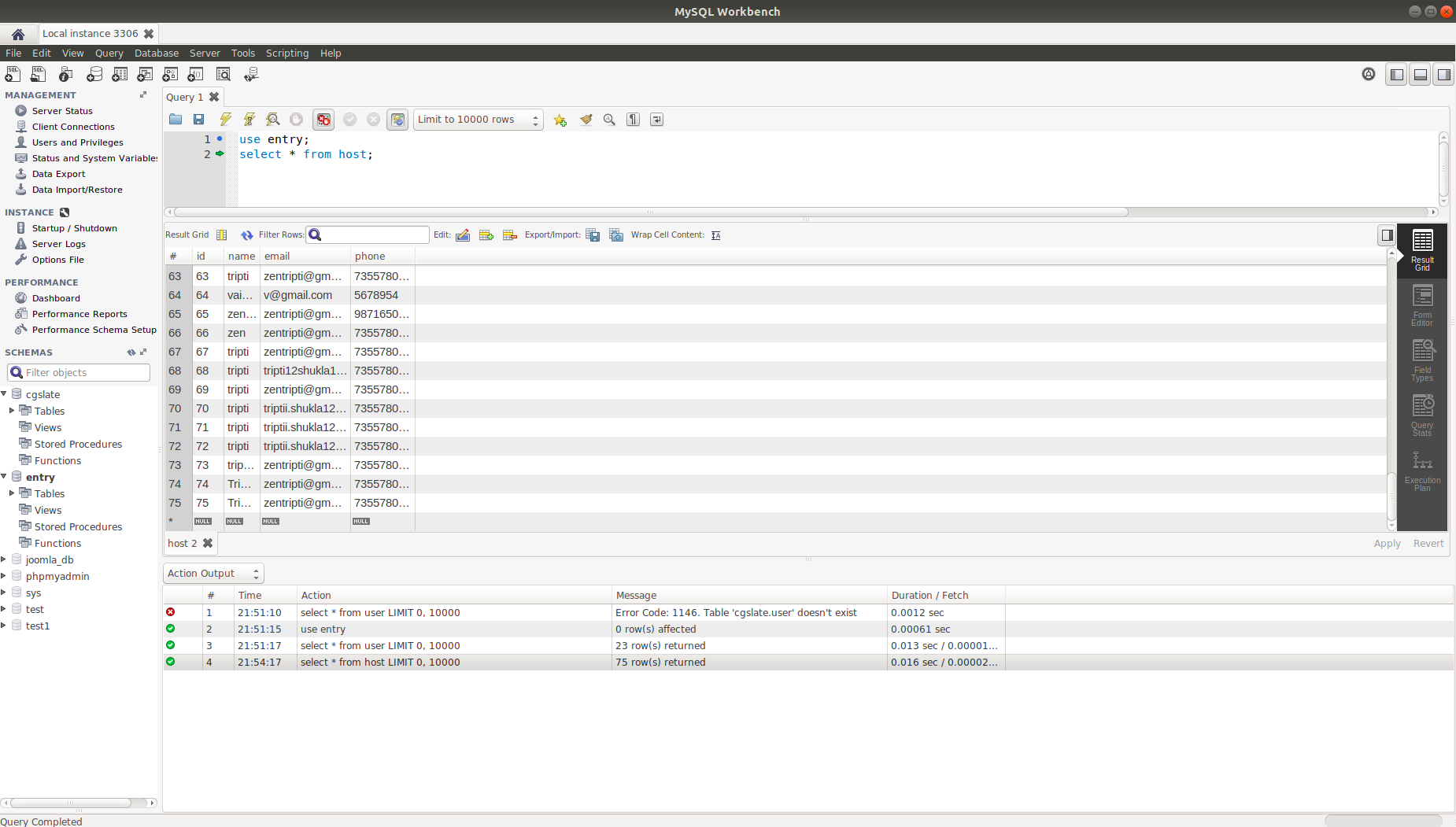
* As soon as the user presses the enter button an API call is made to <http://localhost:5000/api/login> and the user details are inserted into the **user** table in the **entry** database.



* Once the user presses the submit button, the host is expected to switch to the host tab and enter his/her details. As soon as the host presses the submit button, an API call is made to <http://localhost/api/register> and the host details are entered into the **host** table of the **entry** database.

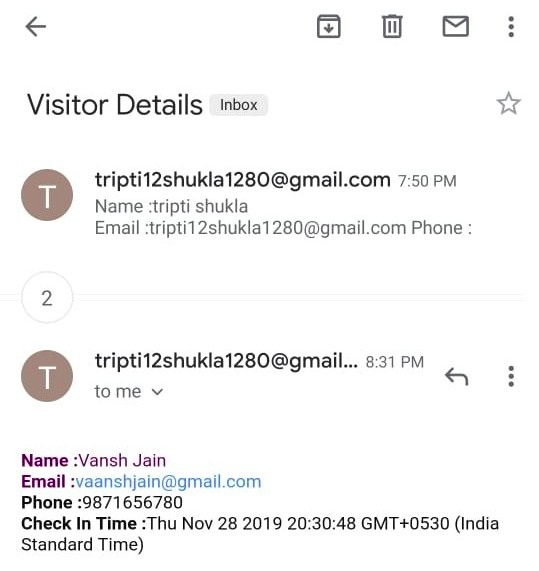


Host Table in entry database:

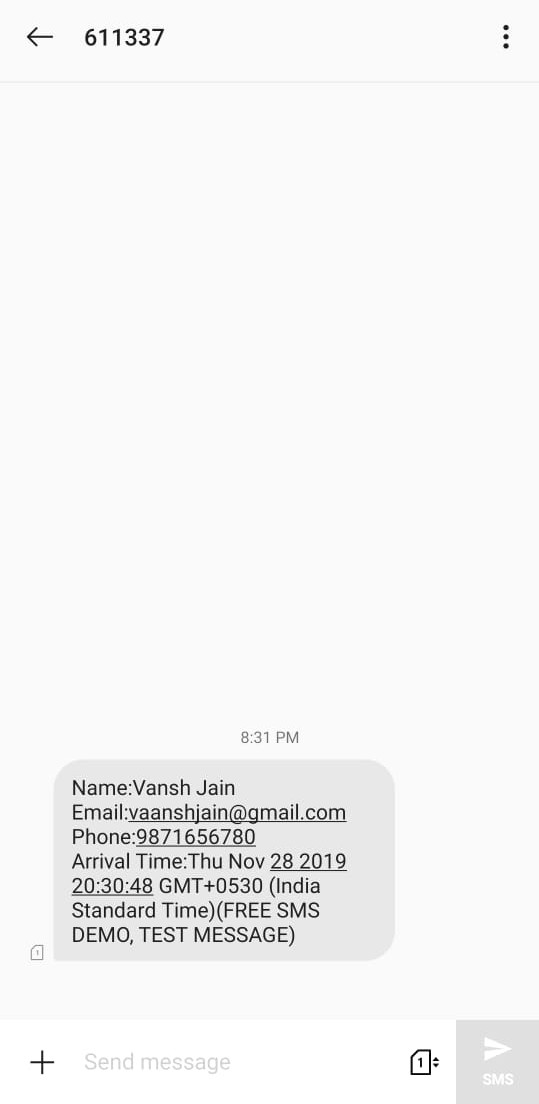


* Once the details have been successfully entered into the table, an email will be sent to the host’s email address that was passed as a request in the API call with the details about the visitor. At the same time an SMS will also be sent to the host’s phone number regarding the details of the visitor.

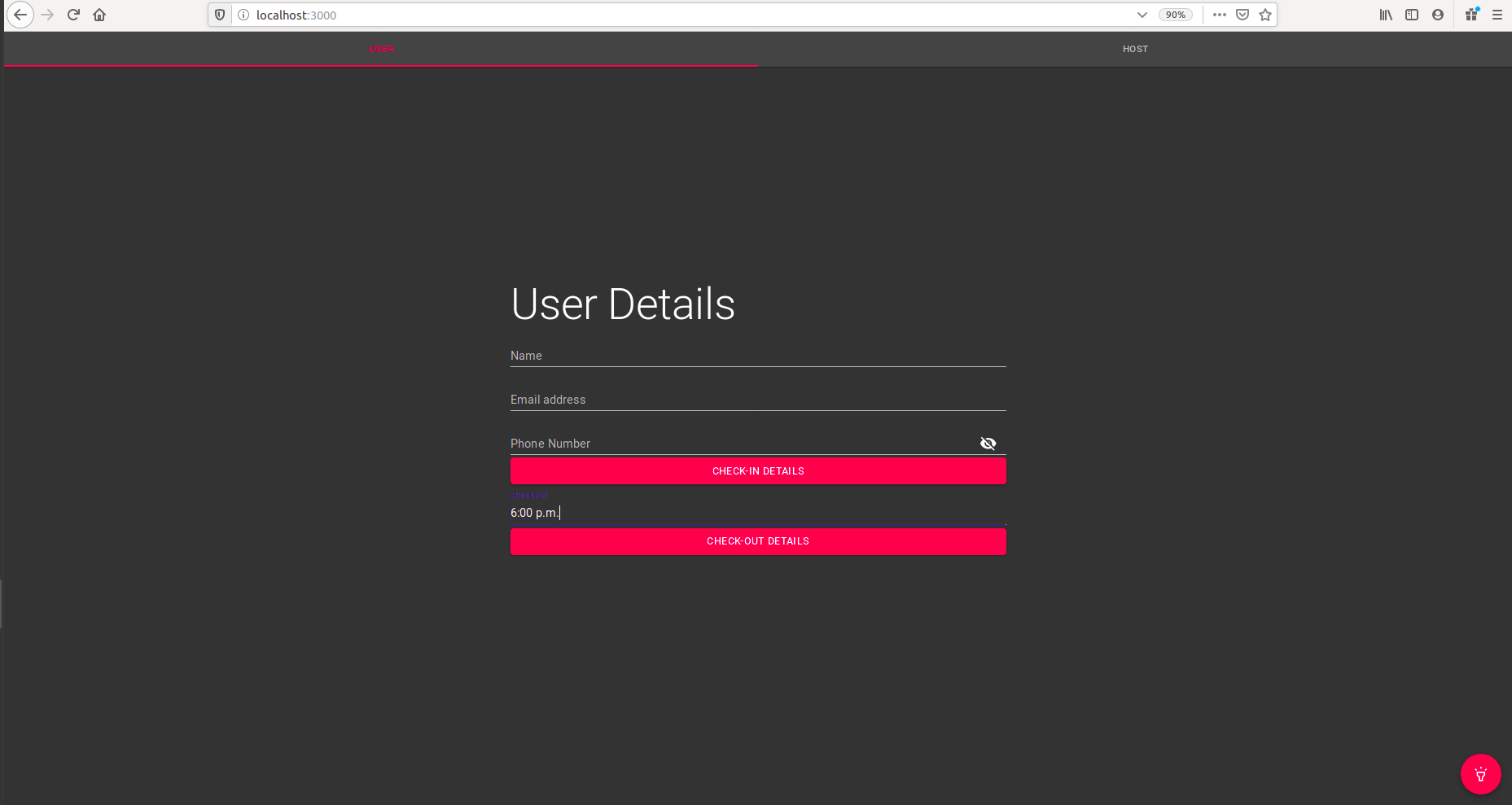
Email sent to the host :



SMS sent to the host:



* Now at the time of checkout the visitor will enter his/her checkout time and press the **checkout details** button.



* As soon as the button is pressed on the user details screen only, an API call will be made to <http://localhost/api/checkout> wherein the checkout time will be updated in the database with the departure time entered by the visitor and a mail will be sent to the visitor with all the details about his visit to the Innovaccer’s office.

Mail sent to the visitor:

