# Database of users, movies and parking

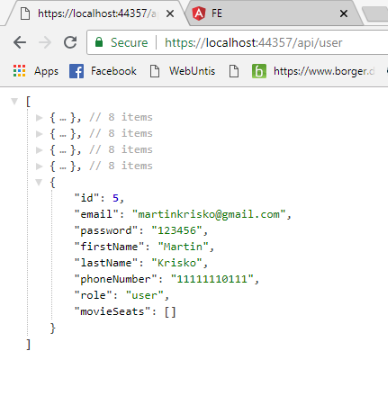
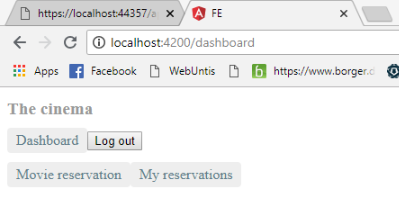
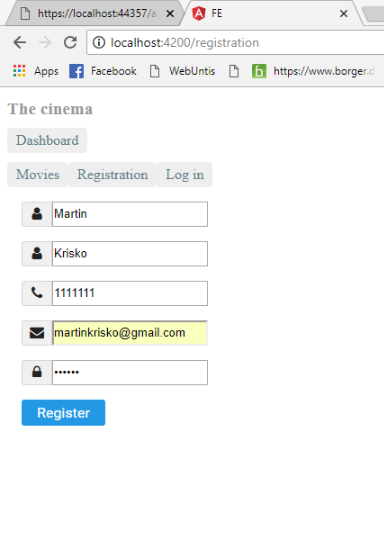
Once we call server API with movie, user and parking endpoint it should show information stored in the database



We are able to see all tables with required information inside therefore we conclude this test to be successful.

# Registration of user

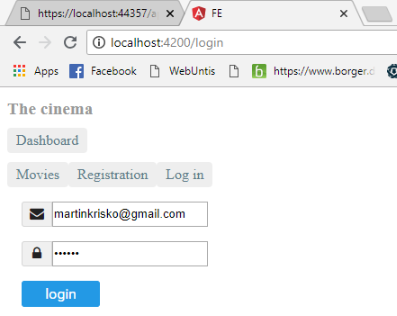
User will click on registration panel and then fill the form in order to be valid. After pressing Register button user should be created in the back end and then logged in at front end.

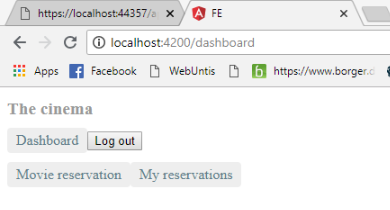


As we can see the operation did what it was supposed to therefore we closed the Registration requirement.

# Login for users

User is able to login. We will fill the user login form on the front end and we should be able to browse our webpage. Also, it should store the user to the local storage so upon refresh he should stay logged in.

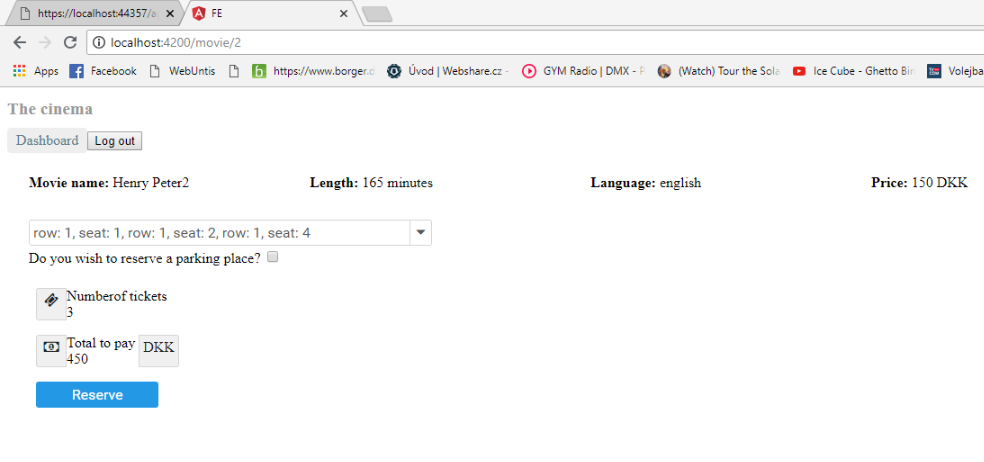




After we fill the login form it logged us in to the application and it created user in local storage, so we can refresh the page or even leave the page and upon return we will still be logged in.

# User can book a movie seat.

User is logged in and can see the movies. He will then pick one of them, which will lead him to detail component. Here he has to pick seats he wants to reserve and then click on Reserve.

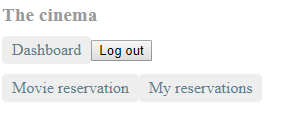
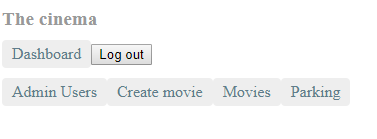




User assigned to a seat with his ID.

# 5. Admin and user

We want to have an admin user who would be able to perform special operations on the web page.

 Admin User

As can be seen on the screenshots above there are 2 types of users stored in the database admin and user. And on picture below the first one is displayed the difference between dashboard of admin and user. Therefore, we consider dis requirement to be fulfilled.

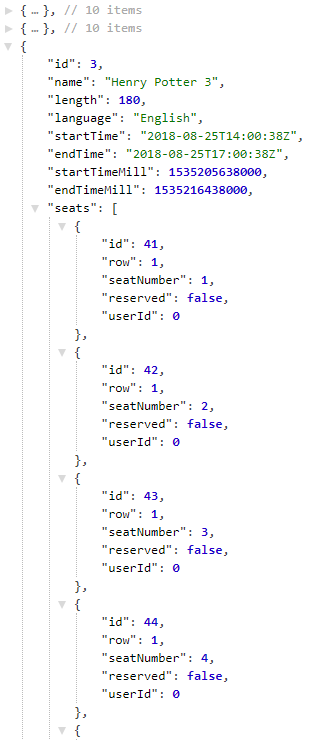
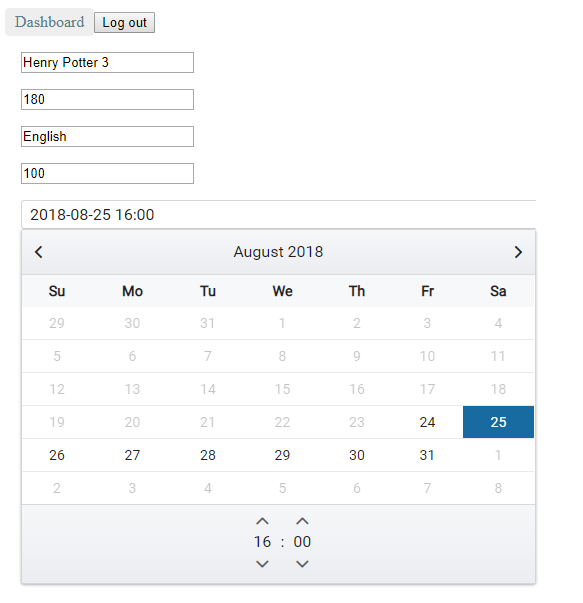
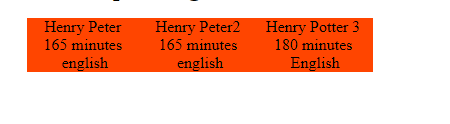
# 6. Price calculation

We expect that the user of the website will be able to see the price for his tickets when reserving the movie seats.

As can be seen on the picture above, if the user selects two seats the number of seats as well as price to pay for the tickets is displayed.

# 7. Admin can create a movie

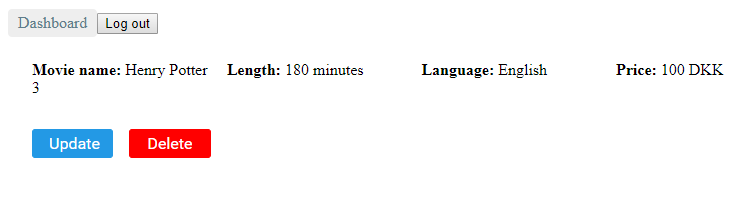
Admin should be able to create a movie once he is registered and fills in the required field.

Create movie form Result database

As can be seen on the pictures above once the admin fills in the form which specifies the movie details and clicks on a button create show, the new movie is created in the database and can be accessed from the frontend.

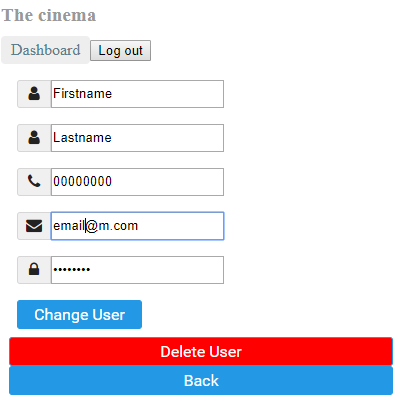
# 8. Delete and update show

Admin should be able to delete and update a movie from the list of movies.

As it is displayed on the pictures above the admin can remove a show from the list of upcoming movies, however he is not able to update a show. Therefore, this requirement is considered not fulfilled.

# 9. Delete and update user

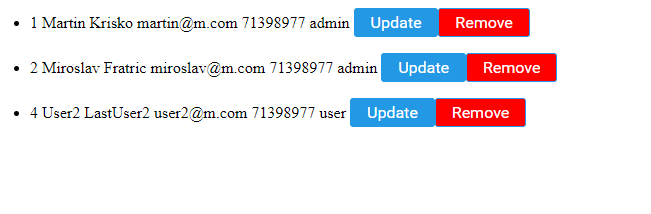
Admin should be able to remove or update e user if needed.

Database before update

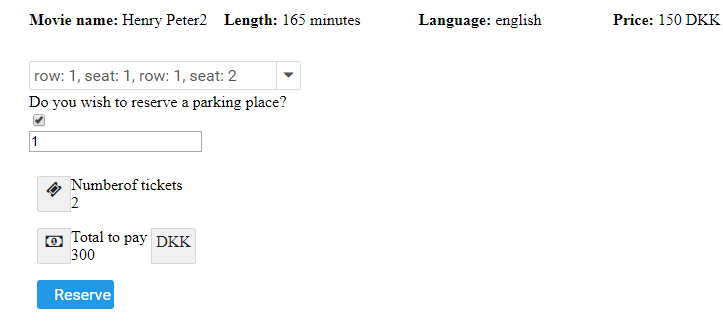
Database after update



Database after delete

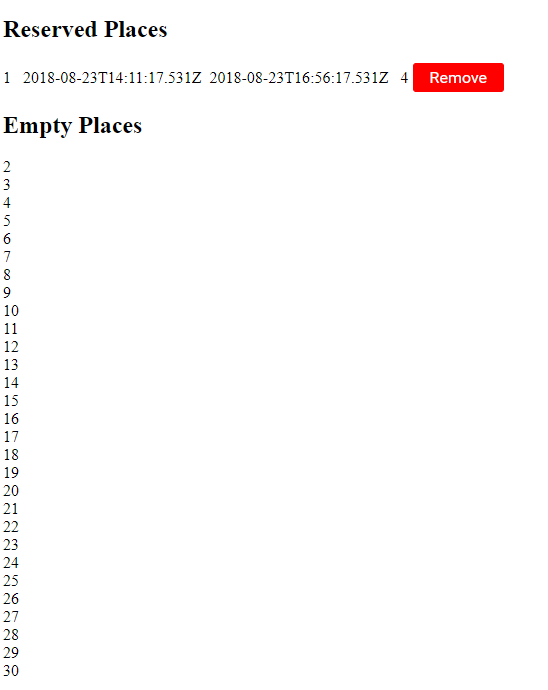
As can be seen on the pictures above once the admin changes the details of a user they are correctly saved in the database. If the admin clicks on the delete user button the user is deleted from the database and from the list of users. This requirement is fulfilled.

# 10. Book parking

User should be able to select if he desires to reserve a parking place as well as number of parking places when reserving a movie ticket.

As can be seen on the pictures above, if a user selects that he desires to reserve a parking place a parking place is assigned to him in the database. This requirement is fulfilled.

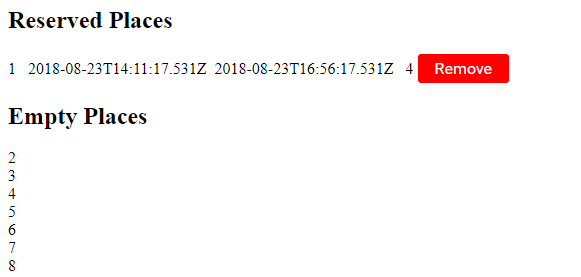
# 11. Overview of parking places

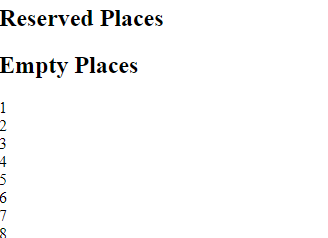
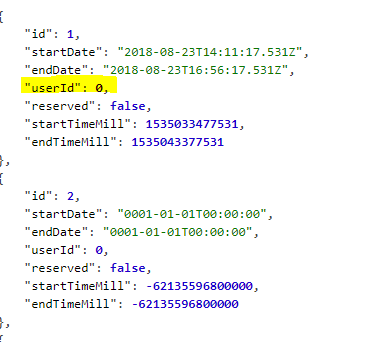
Admin should be able to see the list of reserved and free parking places.

Once admin choses parking from the menu he is shown the list of reserved and free parking places. Requirement fulfilled.

# 12. Remove parking reservation

Admin should be able to remove a parking reservation from the list of reserved parking if there is any.





As can be seen on the pictures above after an admin clicks on delete next to a parking reservation it is delete from the list of reserved places and the user id and reserved field from the database is set to 0 and false respectively. This requirement is considered to be fulfilled.