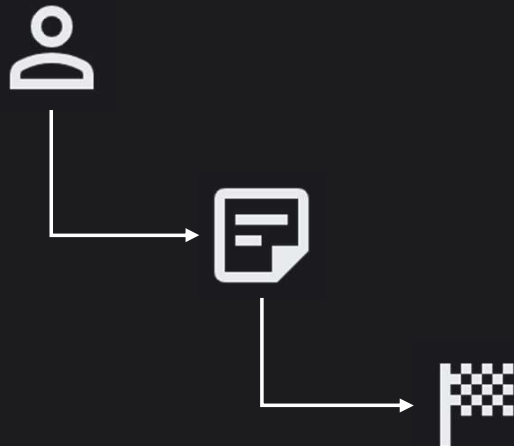# WnCar

A project by

Di Martino Ludovico, Galli Filippo, Leonardi Alessandro, Rigobello Manuel, Scapinello Michele

WaCar

CAR EXPERIENCE MANAGEMENT SYSTEM

WaCar — Home · Circuit List · Insert Car · Insert Circuit · Insert Mapping · Account · Logout

Cars list

Maserati GranTurismo Trofeo
performance
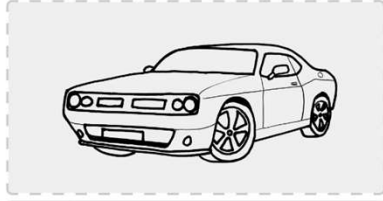
Lamborghini Huracan Tecnica
supercar

Pagani Zonda R
hypercar

WACAR
WaCar offers you a platform to book an experience with your favourite car, on your favourite track! What are you waiting for?!!

WaCar — Home · Circuit List · Car List · Insert Circuit · Insert Mapping · Account · Logout

Add new Car

Brand

Model

Select car type · Add new **car type**

☐ Available

Scegli file · Nessun file selezionato

Description

Max Speed · Horsepower · 0-100 (in seconds)

Submit

WACAR
WaCar offers you a platform to book an experience with your favourite car, on your favourite track! What are you waiting for?!!
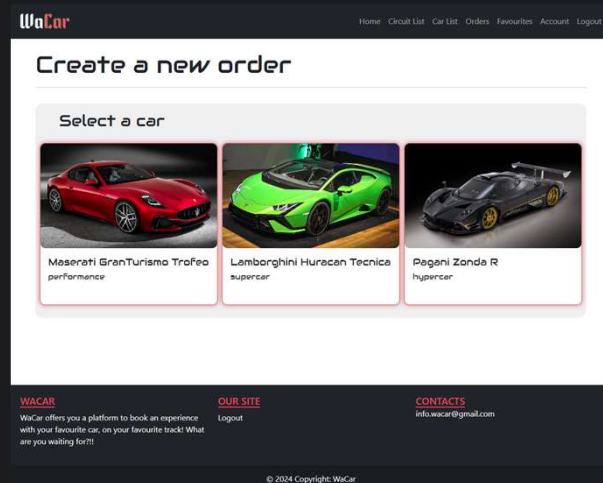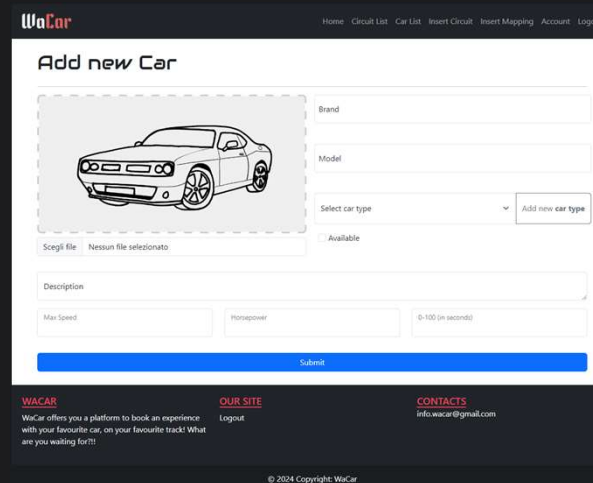
OUR SITE
Logout

CONTACTS
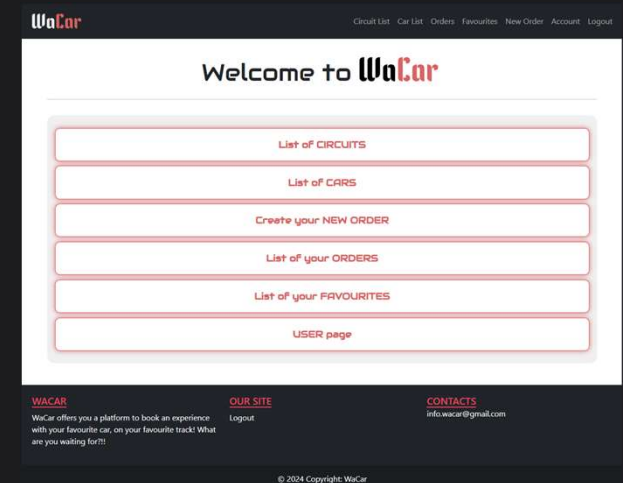info.wacar@gmail.com

© 2024 Copyright: WaCar

Goals

Easier way to order

Easier management of the products

User-friendly & responsive design

Types of User

Basic user
- Create and Manage her/his own orders

Administrator
- Administrator Features
- Cars and Circuits management

REST Resources

URI	DESCRIPTION
/rest/{car|circuit}/	LIST all CARS/CIRCUITS
/rest/user/order/create/{carType}	GET Circuits based on {CarType}
/rest/user/order/create/complete	Create the Order
/rest/user/order/{orderId}	LIST ORDER based on the ID
/rest/user/favourite/{add|delete}	CREATE/DELETE favourites ORDER
/rest/circuit/{circuitName}	GET circuit with name={circuitName}
/rest/car/{carBrand}/{carModal}	GET car with brand={carBrand} and modal={carModal}
/rest/user/favourite/search/{circuitName}/{carBrand}/{carModel}	GET the favourite with the selected values of the logged in user

REST Flow

REQUEST ⟶ RESTDispatcherServlet ⟶ Rest Resource ⟶ DAO

```java
/**
 * Performs the serving logic for listing cars. Retrieves a list of cars from the database
 * using a Data Access Object (DAO) and sends the JSON representation of the list to the response output stream.
 * If successful, sets the HTTP status code to SC_OK (200). If an error occurs during the database access,
 * sets the HTTP status code to SC_INTERNAL_SERVER_ERROR (500) and sends an error message JSON response.
 *
 * @throws IOException  If an error occurs while writing JSON data to the response output stream.
 */
👤 Michele Scapinello +1
@Override
protected void doServe() throws IOException {
    Message m = null;
    List<Car> cars = null;
    try {

        // creates a new DAO for accessing the database and lists the cars
        cars = new ListCarDAO(con).access().getOutputParam();

        if (cars != null) {
            LOGGER.info("Car(s) successfully listed.");
            LOGGER.info(cars.getFirst().getModel());

            res.setStatus(HttpServletResponse.SC_OK);
            new ResourceList<>(cars).toJSON(res.getOutputStream());
        } else {
            LOGGER.error("Fatal error while listing car(s).");

            m = new Message(
                    ErrorCode.UNEXPECTED_ERROR.getErrorMessage(),
                    ErrorCode.UNEXPECTED_ERROR.getErrorCode(),
                     errorDetails: null);

            res.setStatus(ErrorCode.UNEXPECTED_ERROR.getHTTPCode());
            m.toJSON(res.getOutputStream());
        }
```

```java
/**
 * Checks whether the request if for visualizing an {@link Order} and, in case, processes it.
 *
 * @param req the HTTP request.
 * @param res the HTTP response.
 * @return {@code true} if the request was for a {@code Product}; {@code false} otherwise.
 * @throws Exception if any error occurs.
 */
1 usage  👤 manuel_rigobello +1
private boolean processOrder(final HttpServletRequest req, final HttpServletResponse res) throws Exception {
    final String method = req.getMethod();

    String path = req.getRequestURI();
    Message m = null;

    LOGGER.info("I am here");

    // the requested resource was not an order
    if (path.lastIndexOf( str: "user/order") <= 0) {
        LOGGER.info("Return false");
        return false;
    }

    // strip everything until after the /order
    path = path.substring( beginIndex: path.lastIndexOf( str: "order") + 5);
```

Generic Flow

Post/Get

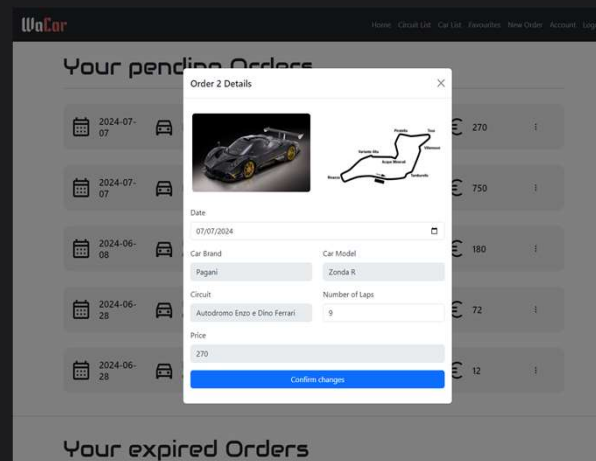Servlet

DAO

JSP

Return and
visualization
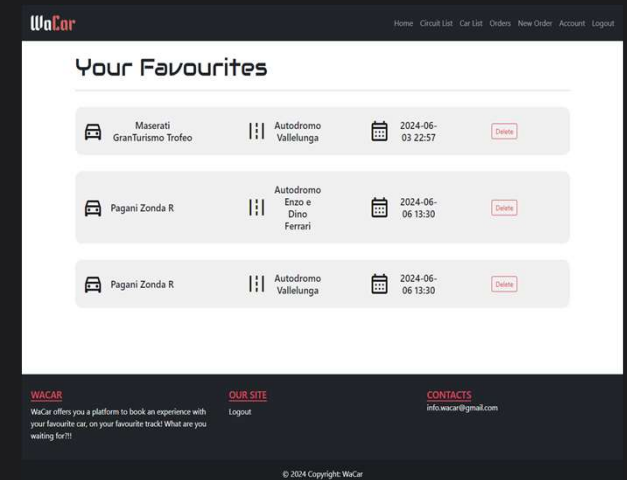
User Features

Create and manage Orders
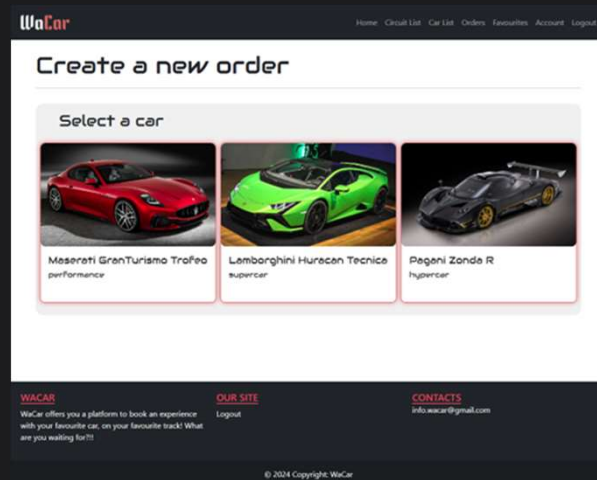
1. Create

2. Modify/Delete

3. Favourites Feature

Order procedure

1. Select a Car → 2. Select a Circuit → 3. Select a Date → 4. Select the Number of Laps
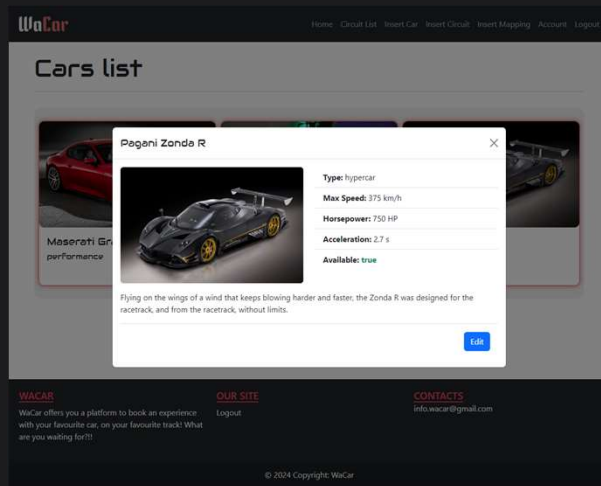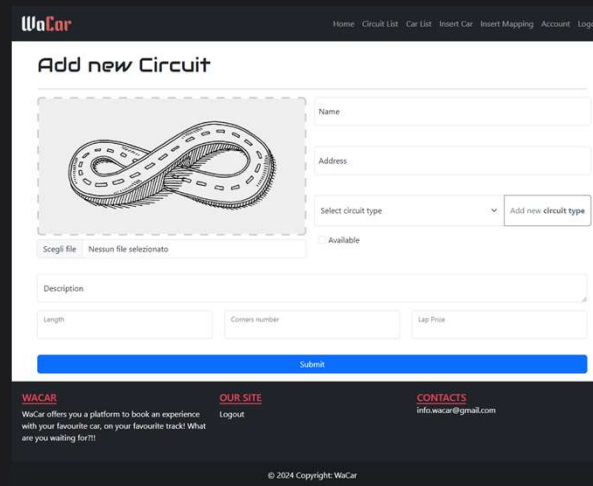
5(b). Add to Favourites

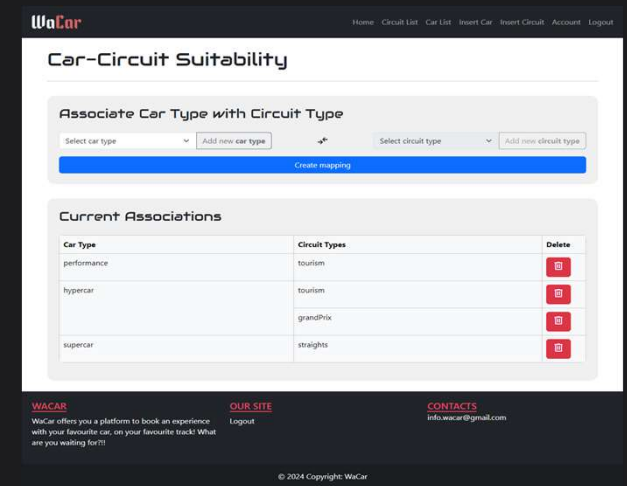5(a). Complete Order
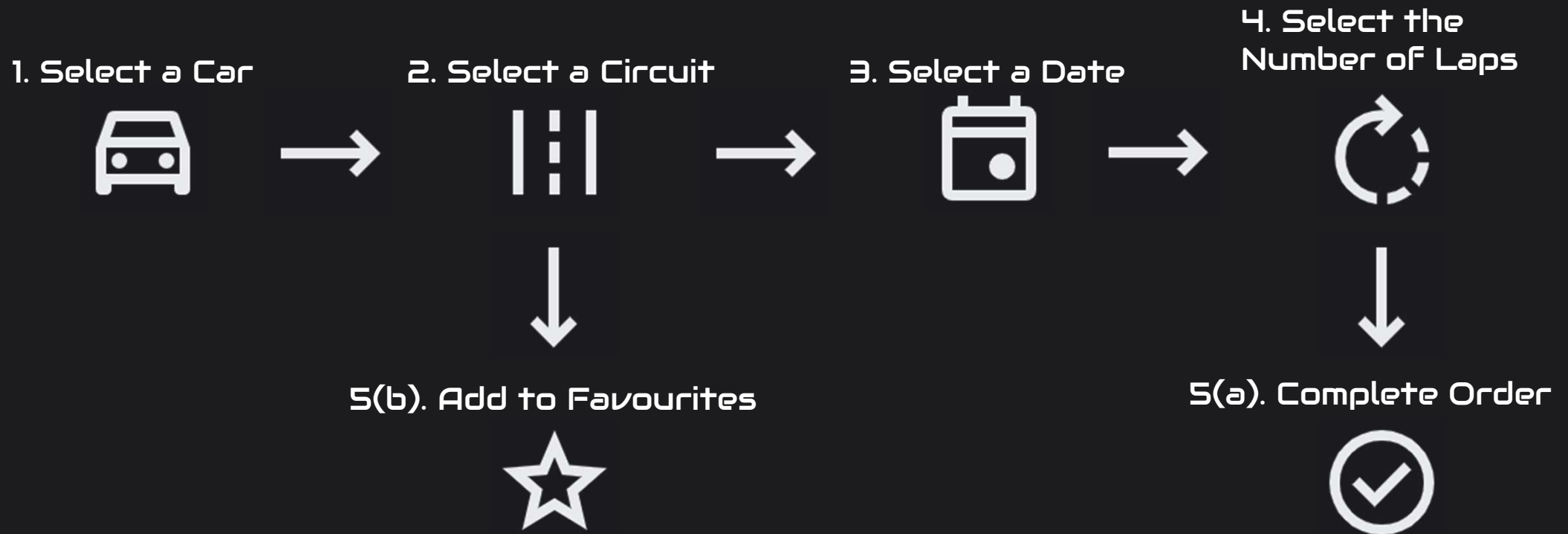
Admin Features

Cars/Circuits management

1. Edit

2. Insert

3. Map

Responsive Design
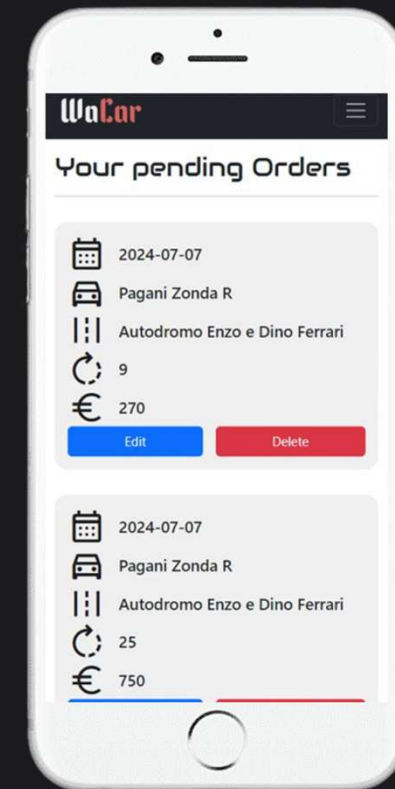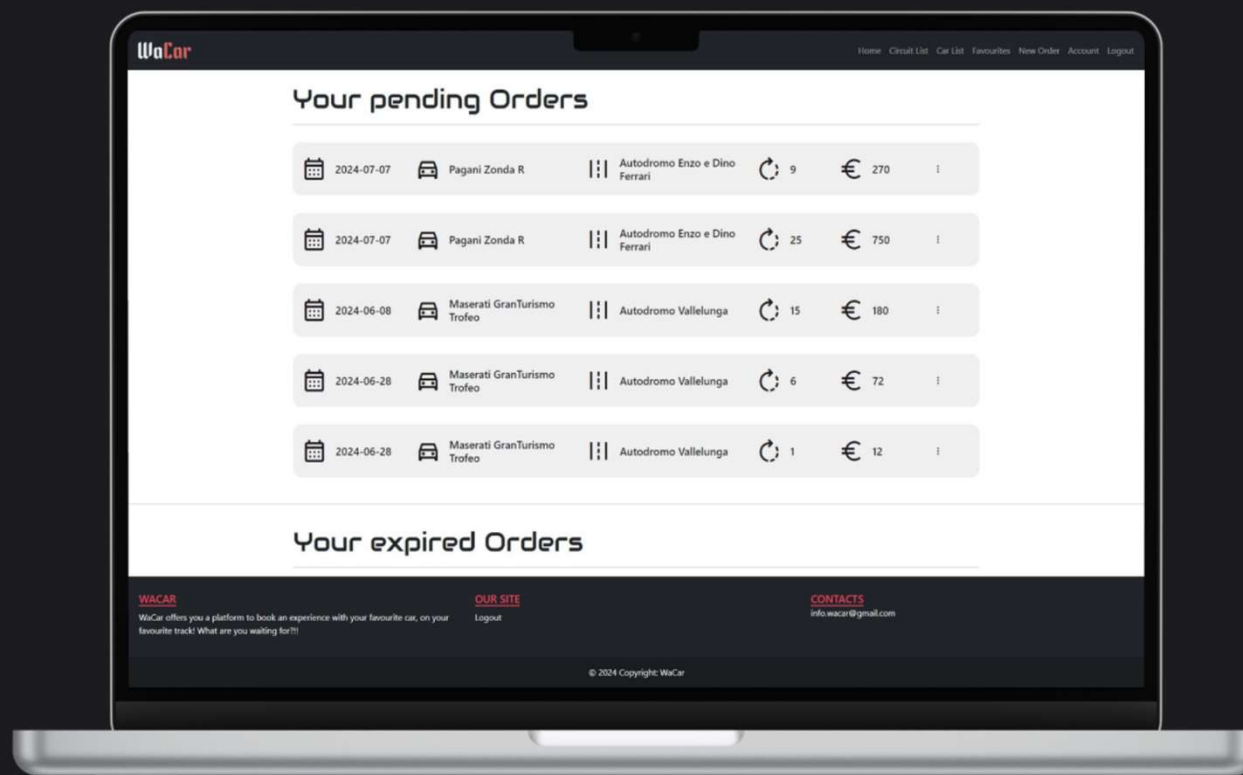
Conclusions

- WaCar satistifies the needs of customers and owners with its simple interface and usage

- The goal of the project has been achieved by providing a solution that facilitates the book of such experiences and the management of the offered products

- There is room for future improvements in the *web* application. Specifically, order management by the admin and the ability to manage order timing could be implemented.

Contributions

- **Di Martino Ludovico**: developed the frontend of the admin section and the order recap modal for the basic user. Created and added the favicon and the logo on all the pages of the website.

- **Galli Filippo**: developed the frontend for the login, the signup and the modify-order modal. Furthermore I have developed the UpdateOrder servlet and the navigation bar

- **Leonardi Alessandro**: developed the frontend of the list favourite page. Contributed to the development of the frontend of the list order page.

- **Rigobello Manuel**: developed the frontend of the list car, list circuit, their modals (with "edit" handling for the admin) and their respective REST. Developed the frontend of the create order page (no recap modal), the REST for getting the circuits based on the carType and the second modal with success/error message (error can arrive from PostgreSQL trigger). Developed the basic user Authentication header with BASIC authorization.

- **Scapinello Michele**: developed the frontend of the list order page and the delete order modal and servlet. Development of the footer and part of the navbar. Preparation of the slides.