Transcript for the video

Hello, we're group 10.

In the subject IDATA2301 & IDATA2306, we have been assigned to make a website. In the project we have been assign with the theme RentalRoulette. This main function for the website it to rent out cars from different providers. Where the providers can rent out car through the site. We have interpreted that we are just a middleman, and don't have a warehouse, but the customer must fetch the vehicle at the rented provider.

Work Methodology we have used is sprint reports and scrum boards. Which we have updated through Microsoft Teams. We have used Github for version control, where we have used our different branch, and if something works, we push it to main branch. Mostly through the project we have sat together, and work either through our own branch or through live share on VSCode. We have tried to work on different projects or field to avoid must use time on merge conflicts etc.

The architecture we have used is a decoupled architecture. Where we have divided frontend and backend into two different folders. Frontend we have used Next Js, where we have uploaded the web through their site and connected it to the domain. At the backend we have used NestJs, Prisma and PostgresSQL. Here we have uploaded backend to the open-stack server provided by NTNU. We also have used JWT Token for authorization for security.

We have chosen the two parted architecture because for later scalability and while we learn more about how to connect backend and frontend.

Design have we kept it clean and simple, where we have used light colours and dark colours. In dark mode this switch. We have used components from Shadcn to maintain a modern style and to have the process most effective.

Here you can see the wireframes. We have tried to not make them as complicated and detailed but kept it simple and clean to keep it easier for us later to program. This is the reason because we haven't coded a lot of websites before, so we aren't sure how hard it is to program a details site. Here you can see the homepage, where you can see the signup button, registration button, and search bar. We also show some cars. Here you can see the login form, where you input email and password. Here is the registration page, where you input the information needed for database later. Last, we have a vehicle page, where you can see all the cars, and a search bar where you can filter where you want to rent, when and what, etc.

Here is a walkthrough about the website, rentalroulette.com. Here is the main page, where you can find an AI generated picture of Ålesund. Here is you will find the ten most popular cars. Here you have the possibility to take a change to get a random car, since the name of the website and theme is rental Roulette. In the header you will find link to home, vehicles and about. In vehicle page, we show all the vehicles, where you also can filter after size, fuel, etc. Here is the about page.

In the footer, we have linked to the Github repository, and beside that we have a link to all the libraries and references we have used on the website.

Here you got the cart and dark mode.

At the login you can create a user, test user.

When you have logged in you can add to favourite and rent cars. You can remove from the cart, or you can clear the whole cart. If you checkout, you can see your orders, and here you can see your favourite.

If we log into an admin user, you will get access to the admin bar in the header. Regular users don't have access to this function. We can see the orders from what have been rented. We can see the order we just made by the test user.

At the product page we could see all the vehicles, the Peugeot is hidden, we could see that with the red mark. We could change that on the right side. We could also add a vehicle. We could show all the users, where we could see the test user has arrived.

If we are on a mobile, we can see the window getting scaled down, and the header will turn into a button at the left side. The vehicle list will scale after how big the screen is.

We run a lighthouse test.

Here is the API documentation. api.rentalroulette.com/api.

Here you can find all the API endpoints and documentation about every endpoint. Here is an example where you could see all the vehicles in database. We have used swagger to check if all the endpoints are working.

Here you could see the database diagram. We have design it to effectively handle the user, orders, and rentals. We can see the user and vehicle have a relation to most of the other tables, which cover the needed part of our website.

Here is the postman tests we have created. We start with showing the environment file. The first test is login, where we have used the admin email and password. If you use the wrong credentials, you will get invalid credentials error. Here you can get all users if you have the bearer token. If you try to get all the users without the token, you will get an "unauthorized" error. Here we can create a test user.
For vehicles you can get all the vehicles. Using the post request, you can create a vehicle, which in this case gets assigned to id 43. If we then put this id into the patch method, we can update the vehicle. Lastly, you can delete a vehicle with the vehicle id. Here you can get all the orders, and if you are not authorized you will get an "unauthorized" error.

The extras:

Try your luck, where you get a random car.

Ten most rented car.

Dark mode.

Advanced search filter.

Extra function in admin page, overview of orders, add vehicle, update vehicles, delete vehicles.

Same with user, where you could edit and delete.

All logged in user can update their profiles.

Reflection:

At the reflection we can start with the downside, and that is the car can’t be rented for a period of time, and then they will not show for users. We have implemented in the database but didn’t have the time to get it to work.

We have learned a lot about frontend and backend programming, and how to make a fullstack web application. We have learned a lot about web design and usability.

We also have learned how to connect frontend and backend, and it not always so easy to connect them together at first. We also have learned to uploading to a different server, apart from localhost, and it not always it works on first try.

We also have learn working in teams, and figured out what works and don’t work. Also, when it comes to design of the website, how the input from other users is usefull, because it is in the end their the one who is going to use it.

Thank you for listening at Group 10 and have a great summer!