

FINAL REPORT – HCI

EASY RENT

**Made by
Brstilo, Filip
Jurić-Pešić, Mijo**

Note: This report is made by combining previous 6 reports and making 7th final report

Figma essentials

In this report, we will explain how we made this phase of the project. Here is the table of contents for easier navigation.

Motivation

Creating hero section

 Creating navbar (using assets and instances)

 Responsive design

Conclusion

Motivation

Using Figma, design tool used to create user experience (UX) prototypes, we have made simple yet responsive prototype that will be useful for our big project this semester. Figma is a powerful tool that allows you to add UX components in just a few clicks. It also allows you to create responsive design and to collaborate with multiple people on the same project.

Before you write a single line of code, it is always a good thing to make a prototype to show to your clients/customers so they can give you useful feedback and suggest for any redesigns.

Creating hero section

Creating navbar (using assets and instances)

First, we start by creating the new frame that is going to be a base for our project. Next, we created second frame that we will use to create our logo, navbar items and so on. In that frame, we will store everything that will be used on our page. It is good thing to do it like that because we may need a same button, navbar item or our logo on multiple places. Picture below shows You how our 'template frame' looks like.



Here we can see the logo, navbar that we initially created (home home home home) and NAV ITEM that is used as active (bold) or default component. By making an update to any of those components, we apply all the changes that we made to the parent element to all instances of the given component. Using that we created navbar that can change its content, active or default state and change order of the elements without having to manually adjust for spacings.

Responsive design

After creating navbar, we added hero section just to test responsivity of our prototype. Hero section consists of:

- Hero title
- Dummy text
- Rectangle that represents an image

After all of that we get a prototype that looks something like this:



After that we use constrains that 'anchor' certain object or group of objects to a side of our page. To do that we:

1. Select object or group of object
2. Select Position tab
3. Select Constrains box
4. Choose on which axis we want the constrains

Apart from this report and description, there is also a short video that show responsivity of our design.

Conclusion

We used Figma to created simple yet powerful template and we learned concepts such as:

- How to use Figma
- UI/UX prototype

- Assets and instances
- Responsive design

Those concepts will be important in our big final project.

User personas and information architecture

In this report, we will explain how we made this task. Here is the table of contents for easier navigation.

[Motivation](#)

[Introduction](#)

[Creating User Personas](#)

[Analyzing our personas](#)

[Creating AI images for given personas](#)

[Information Architecture](#)

[Card Sorting Method - results by personas](#)

[Sarah Lee](#)

[Robert Carter](#)

[Javier Martinez](#)

[Final Results of Card Sorting Method](#)

[Sitemap](#)

[Conclusion](#)

Motivation

When creating an application or any product, it's crucial to remember that you're not designing for yourself but for real-life users with diverse needs and varying levels of IT knowledge. The purpose of this exercise is to learn how to think from a user's perspective and adapt our design accordingly. Understanding user personas is a crucial step in this process.

User personas are fictional representations of your target audience, based on real data and research. They help you visualize and empathize with your users, allowing you to design products that truly meet their needs and expectations. By creating detailed user personas, we can better understand our users' goals, behaviors, and pain points, which in turn guides our design decisions and helps us create more user-centered products.

Parameters and all other key points will be explained in future paragraphs.

Introduction

To make this exercise possible, we will make use of ChatGPT to generate 3 user personas that are relevant to given application. What we will be making as our project is Rent a Car web application.

The hardest part of testing the page is collecting data and feedback from users. This process is time and resource-consuming in the lifecycle of a web application. It's crucial that user personas are relevant to our project and cover all key demographics and user types. By creating comprehensive personas, we can:

- Gain insights into user behavior and preferences
- Identify potential pain points in the user experience
- Tailor our design decisions to meet specific user needs
- Ensure our application caters to a diverse range of users

Creating User Personas

We figured out that our users will be from ages 18 to 80, so that is a large range of people whose needs and abilities differ. For greater variety, we asked ChatGPT to give us 4 user personas from different demographic groups and to list some of their personal information like their age, marital status, number of children, and a few more details of its choosing.

Here is what ChatGPT came up with:

Persona 1: Young Professional

- **Name:** Sarah Lee
- **Age:** 25
- **Relationship Status:** Single
- **Number of Children:** None
- **Occupation:** Marketing Analyst
- **Location:** New York, NY
- **Education:** Bachelor's Degree in Business
- **Income:** \$50,000/year
- **Favorite Activity:** Going to concerts, traveling with friends, and exploring new restaurants.
- **Goals:** Needs affordable, short-term car rentals for weekend trips outside the city.
- **Pain Points:** High rental costs in urban areas and limited parking availability.
- **Technology Comfort:** High – prefers mobile apps and digital payments.

Persona 2: Retired Senior

- **Name:** Robert Carter
- **Age:** 68
- **Relationship Status:** Married
- **Number of Children:** 3 (all grown)
- **Occupation:** Retired Teacher
- **Location:** Phoenix, AZ
- **Education:** Master's Degree in Education
- **Income:** Fixed retirement income
- **Favorite Activity:** Golfing, gardening, and visiting grandchildren.
- **Goals:** Needs a comfortable and reliable car for occasional long trips and family visits.
- **Pain Points:** Prefers in-person services and may struggle with overly complex booking systems.
- **Technology Comfort:** Low – prefers simple interfaces and customer service assistance.

Persona 3: College Student

- **Name:** Javier Martinez
- **Age:** 21
- **Relationship Status:** In a relationship
- **Number of Children:** None
- **Occupation:** Full-time Student (Engineering)
- **Location:** Los Angeles, CA
- **Education:** Pursuing a Bachelor's Degree
- **Income:** Part-time job (\$15,000/year)
- **Favorite Activity:** Gaming, skateboarding, and going to tech meetups.
- **Goals:** Needs affordable and flexible car rentals for short-term use, especially during holidays or weekends.
- **Pain Points:** Budget constraints and additional fees (insurance, age-related fees) for younger drivers.
- **Technology Comfort:** High – prefers apps with social media integration and mobile-first design.

Persona 4: Single Parent

- **Name:** Lisa Nguyen
- **Age:** 34
- **Relationship Status:** Divorced
- **Number of Children:** 2
- **Occupation:** Nurse
- **Location:** Dallas, TX
- **Education:** Nursing Degree
- **Income:** \$60,000/year
- **Favorite Activity:** Taking her kids to sports practices, family outings, and fitness classes.
- **Goals:** Needs an affordable, spacious vehicle for family trips with child-friendly features (like car seats and safety features).
- **Pain Points:** Balancing affordability with quality, ensuring safety features for children, and finding flexible booking times around a busy schedule.
- **Technology Comfort:** Moderate – prefers mobile apps with clear pricing and easy-to-navigate user interfaces.

Personas 1 and 4 are quite similar, and since the requirements of the task specify using 3 personas, we will eliminate Persona 4 and stick with the first three.

Analyzing our personas

We have 3 personas in total: Young Professional, Retired Senior, and College Student. They all have their strengths and weaknesses that we will have to take into account when designing the application. Taking that into account, we can start with the next phase of our project, and that is generating AI images for our personas.

Creating AI images for given personas

Creating images is not difficult using today's tools, but it is an important step in our development. People are visual beings and they connect over images, so creating them is a good idea. Also, developers create connections as well. When you say: "*I develop this for x who is going to use this for y*", your application is not

going to be user-oriented. When you say: "*I will make this good for Sarah so she can travel safely from A to B*", then your application has prosperity.

For creating images, we used tool called Canva. In our opinion, it creates more human like images. Here is a comparison between ChatGPT and Canva when we gave them description of Sarah Lee:

- **Name:** Sarah Lee
- **Age:** 25
- **Relationship Status:** Single
- **Number of Children:** None
- **Occupation:** Marketing Analyst
- **Location:** New York, NY
- **Education:** Bachelor's Degree in Business
- **Income:** \$50,000/year
- **Favorite Activity:** Going to concerts, traveling with friends, and exploring new restaurants.
- **Goals:** Needs affordable, short-term car rentals for weekend trips outside the city.
- **Pain Points:** High rental costs in urban areas and limited parking availability.
- **Technology Comfort:** High – prefers mobile apps and digital payments.

ChatGPT (left) & Canva (right)



Seeing these differences, You can see why we choose Canva for our project. Here are all of our 3 personas from left to right: Sarah Lee (29), Robert Carter (68) and Javier Martinez (21)



Information Architecture

Using ChatGPT we found out what Information Architecture is:

The screenshot shows a dark-themed ChatGPT interface. A user question is displayed in a rounded rectangle at the top: "What is Information Architecture? Can you explain that to us in couple of short paragraphs?" Below it, a response from ChatGPT begins with a circular profile icon containing a swirl symbol. The text explains that Information Architecture (IA) refers to the way content and data are structured, organized, and presented within a digital system, such as a website or application. It involves creating a logical, intuitive layout that helps users find information quickly and efficiently. IA focuses on labeling, categorizing, and mapping out navigation paths, ensuring that users can easily understand how to interact with the content. The response continues by stating that good IA enhances user experience by reducing cognitive load, making it easier for users to locate what they need without confusion. It involves techniques like card sorting, user flow analysis, and content hierarchy to align the structure with user needs and behaviors. Ultimately, IA aims to create a seamless, user-friendly digital environment. At the bottom of the message, there are several small, light-colored icons: a double arrow, a square, a downward arrow, a question mark, a gear, and a checkmark.

Based on our understanding of Information Architecture (IA), we now realize the importance of focusing on the needs of our users for our Rent a Car web application. By analyzing their needs, we can determine what information they will find important and useful, and what can be "hidden away" or removed entirely. This approach will help us create a more efficient and user-friendly experience, ensuring that key information is easily accessible while avoiding clutter and unnecessary features.

Next, we asked ChatGPT to create 20 most topics and tags we can use in our web application. Then we will ask the to simulate for all personas ranking of each topic and tag in descending order and then looking by all simulations to combine all of them into one result. Here are 20 topics he came up with:

- **Vehicle Categories** – Organize by car types (SUVs, sedans, luxury, economy, etc.).
- **Booking Process** – Step-by-step guide to booking a vehicle.
- **Rental Rates** – Pricing breakdowns by car type, rental duration, and location.

- **Insurance Options** – Information on available coverage for renters.
- **Pick-up & Drop-off Locations** – Maps and addresses for rental locations.
- **User Account Management** – Creating, editing, and managing user profiles.
- **Payment Options** – Supported payment methods (credit cards, PayPal, etc.).
- **Loyalty Programs** – Rewards or discounts for frequent users.
- **Terms & Conditions** – Policies around cancellations, deposits, etc.
- **Fuel Policy** – Information on refueling requirements for rented vehicles.
- **Customer Support** – Contact information and FAQs.
- **Vehicle Availability** – Real-time car availability by location.
- **User Reviews** – Customer feedback on vehicles and services.
- **Safety Features** – Information on vehicle safety and additional equipment (child seats, GPS, etc.).
- **Special Offers** – Promotions, discounts, and seasonal deals.
- **Driver Requirements** – Age, license, and additional driver policies.
- **Mobile App Features** – Integration of mobile booking and user services.
- **Damage Policy** – Information on damages, claims, and deductible.
- **Vehicle Upgrades** – Options for upgrading to premium models or features.
- **Cancellation & Refunds** – Rules regarding cancellations and refunds.

What we did above is called card sorting experiment, where You ask x number of people to arrange certain categories or topics into order they find most logical and **important** to them. By using these results, we will know in which direction we have to go when designing our application.

We asked ChatGPT to do card sorting method by arranging topics into 4 categories:

1. Must Have
2. Good to Have
3. Not Important but Useful

4. Not Needed

Card Sorting Method - results by personas

Not to make this report too long, we will write down only what our personas found out to be in Must Have category.

Sarah Lee

- **Booking Process** – Step-by-step guide to booking a vehicle.
- **Rental Rates** – Pricing breakdowns by car type, rental duration, and location.
- **Vehicle Categories** – Organize by car types (SUVs, sedans, luxury, economy, etc.).
- **Payment Options** – Supported payment methods (credit cards, PayPal, etc.).
- **Vehicle Availability** – Real-time car availability by location.
- **User Account Management** – Creating, editing, and managing user profiles.

Robert Carter

- **Booking Process** – Step-by-step guide to booking a vehicle.
- **Vehicle Categories** – Organize by car types (SUVs, sedans, luxury, economy, etc.).
- **Rental Rates** – Pricing breakdowns by car type, rental duration, and location.
- **Pick-up & Drop-off Locations** – Maps and addresses for rental locations.
- **User Account Management** – Creating, editing, and managing user profiles.

Javier Martinez

- **Booking Process** – Step-by-step guide to booking a vehicle.
- **Rental Rates** – Pricing breakdowns by car type, rental duration, and location.
- **Vehicle Categories** – Organize by car types (SUVs, sedans, luxury, economy, etc.).
- **Payment Options** – Supported payment methods (credit cards, PayPal, etc.).

- **User Account Management** – Creating, editing, and managing user profiles.
- **Vehicle Availability** – Real-time car availability by location.

Final Results of Card Sorting Method

After ChatGPT gave us these results, we asked him what are “Must Haves” in our web application taking into consideration responses from our personas. Here is what he came up with:

Must Have

1. **Booking Process** – Step-by-step guide to booking a vehicle.
2. **Rental Rates** – Pricing breakdowns by car type, rental duration, and location.
3. **Vehicle Categories** – Organize by car types (SUVs, sedans, luxury, economy, etc.).
4. **Payment Options** – Supported payment methods (credit cards, PayPal, etc.).
5. **User Account Management** – Creating, editing, and managing user profiles.
6. **Vehicle Availability** – Real-time car availability by location.
7. **Pick-up & Drop-off Locations** – Maps and addresses for rental locations.

Good to Have

1. **Insurance Options** – Information on available coverage for renters.
2. **Customer Support** – Contact information and FAQs.
3. **Loyalty Programs** – Rewards or discounts for frequent users.
4. **Special Offers** – Promotions, discounts, and seasonal deals.
5. **Mobile App Features** – Integration of mobile booking and user services.
6. **Safety Features** – Information on vehicle safety and additional equipment (child seats, GPS, etc.).
7. **Driver Requirements** – Age, license, and additional driver policies.

Not Important but Useful

1. **User Reviews** – Customer feedback on vehicles and services.

2. **Fuel Policy** – Information on refueling requirements for rented vehicles.
3. **Damage Policy** – Information on damages, claims, and deductible.
4. **Vehicle Upgrades** – Options for upgrading to premium models or features.

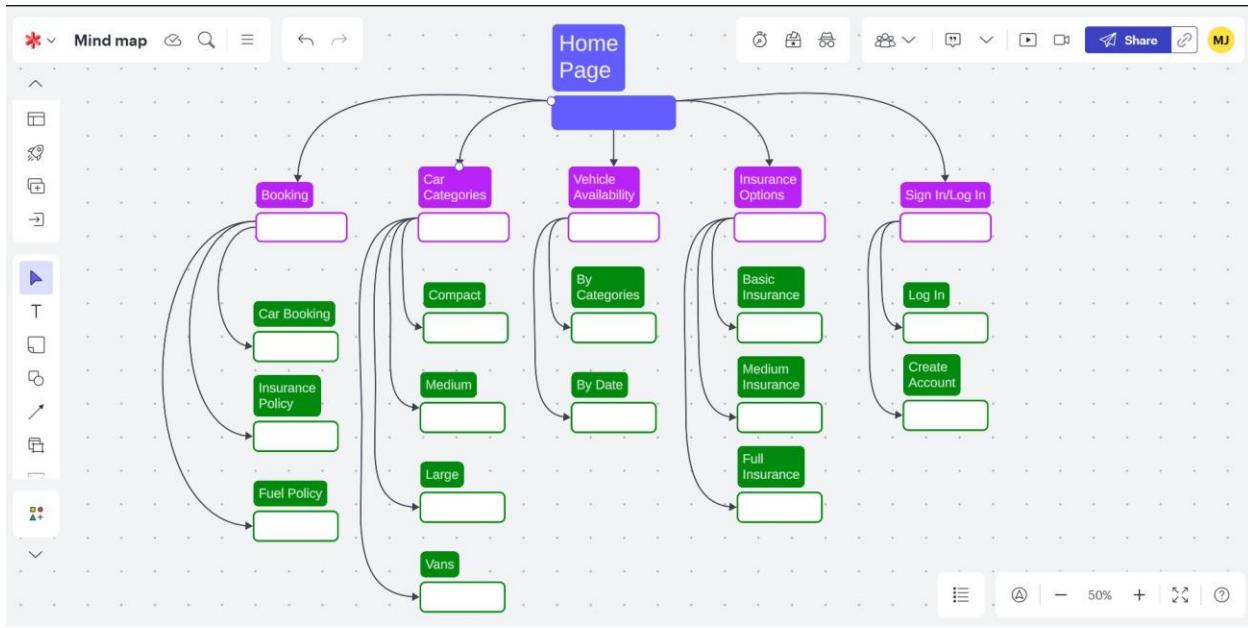
Not Needed

1. **Terms & Conditions** – Policies around cancellations, deposits, etc.
2. **Cancellation & Refunds** – Rules regarding cancellations and refunds.
3. **Real-time Car Availability** – For some personas, this may not be a priority but is still relevant.
4. **Additional Support Resources** – More detailed FAQs or guides that may not be necessary for all users.

Using these results, we know where to go with our design. We will put most important topics at the top and navigation bar for easy access and things like Terms & Conditions at the bottom of the page or in some menu where it is not “in the way”.

Sitemap

The final task in this phase of the project is to create a Sitemap. A sitemap is a structured outline or visual representation of a website's content and organization. Our goal is to create a detailed sitemap for our application, which will be important for future implementations of our webpage and will enhance user experience. We'll use the results provided by ChatGPT and the tool [LucidChart](#) to create a comprehensive sitemap of our web application. We've included five key categories (excluding Home): Booking, Car Categories, Vehicle Availability, Insurance Options, and Sign up/Log in. The subcategories are visible in the image below.



We will use this image as a reference in our future tasks. Of course, if it needs any improvements during the implementation phase of the project, changes will be made. Of course, Professor, Your feedback here is highly appreciated.

Conclusion

During this task we learned a lot about human impact not just in web and application design but in everything else. Humans as visual and feeling full beings need compassion to make the best out of anything. If we as a developers design things to be user oriented, users will have better experience using your product and by listening to their feedback you can push the design to its limits.

From the technical aspect we learned about:

- User personas
- Making user connect with the personas
- Information Architecture
- Sitemap
- ...

Next.js - Deploying Application

Introduction

In this report, we will explain how we made this task. Here is the table of contents for easier navigation.

[Introduction](#)

[Motivation](#)

[Sitemap and establishing routes](#)

[Establishing routes](#)

[Deploying](#)

[Final Product, so far...](#)

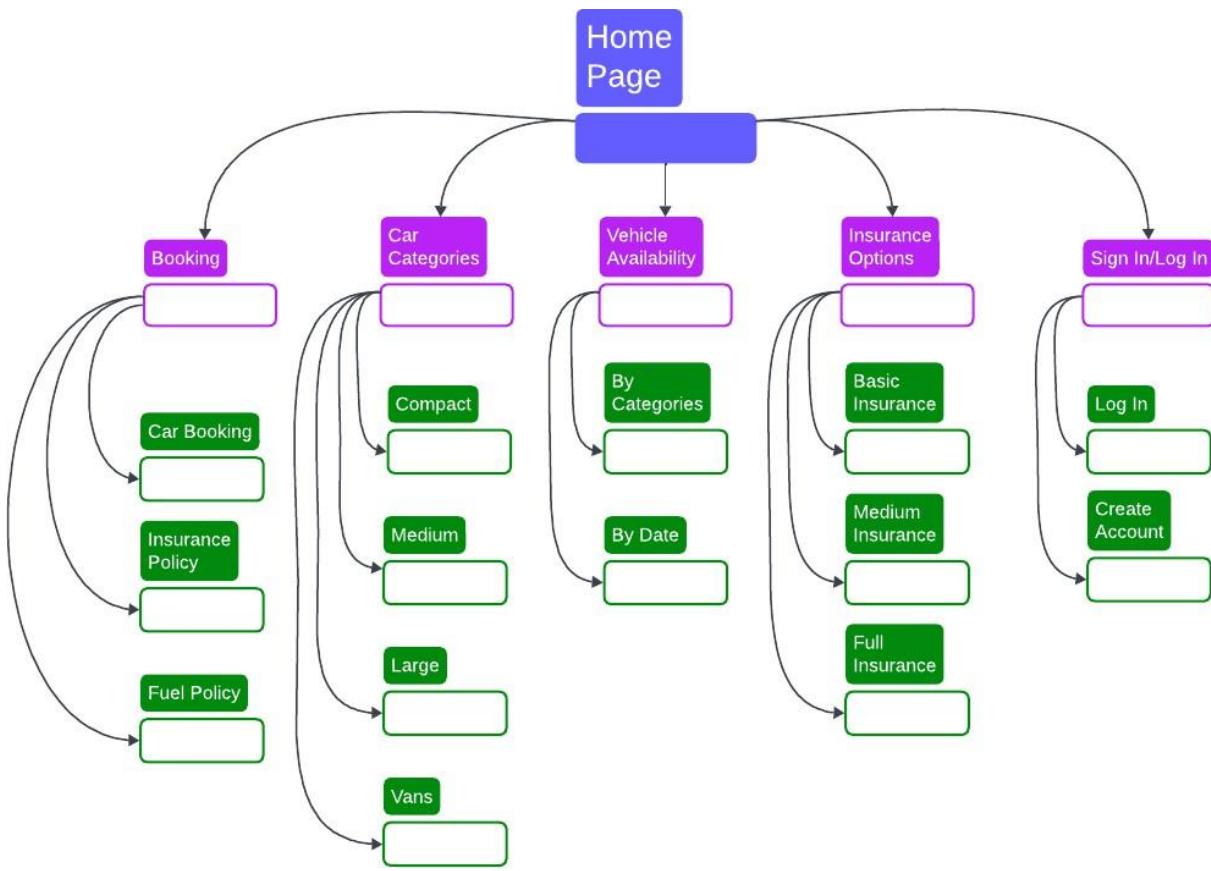
[Conclusion](#)

Motivation

To achieve the best user experience, your web application must be designed thoughtfully. Throughout this exercise, we will strive to accomplish this goal. Remember that users will spend most of their time on other people's pages, so it's important to create something familiar and proven to work, rather than reinventing the wheel. That's the way we will try to design the page-

Sitemap and establishing routes

If you need a reminder, here is a sitemap we created in last exercise:



In our main navigation bar we will have 6 options/navigation items:

1. Home
2. Booking
3. Car Categories
4. Vehicle Availability
5. Insurance Options
6. Sign In/Log In

According to the sitemap, we will also create navigation for each main navigation item, so for example in Car Categories we will have:

1. Compact
2. Medium
3. Large

4. Vans

If later on we figure out these names are not the most brilliant thing ever we will change those.

Establishing routes

Now for the technical bit, we create in each folder of main navbar items these folders

- _components
- sub navigation items folders
- layout.tsx

In _components we make navigation.tsx in which we just write down paths for that navigation items. In sub navigation folders we just put the title of the page for now since we will have task just about that and we use layout.tsx just to import and display corresponding navigation.

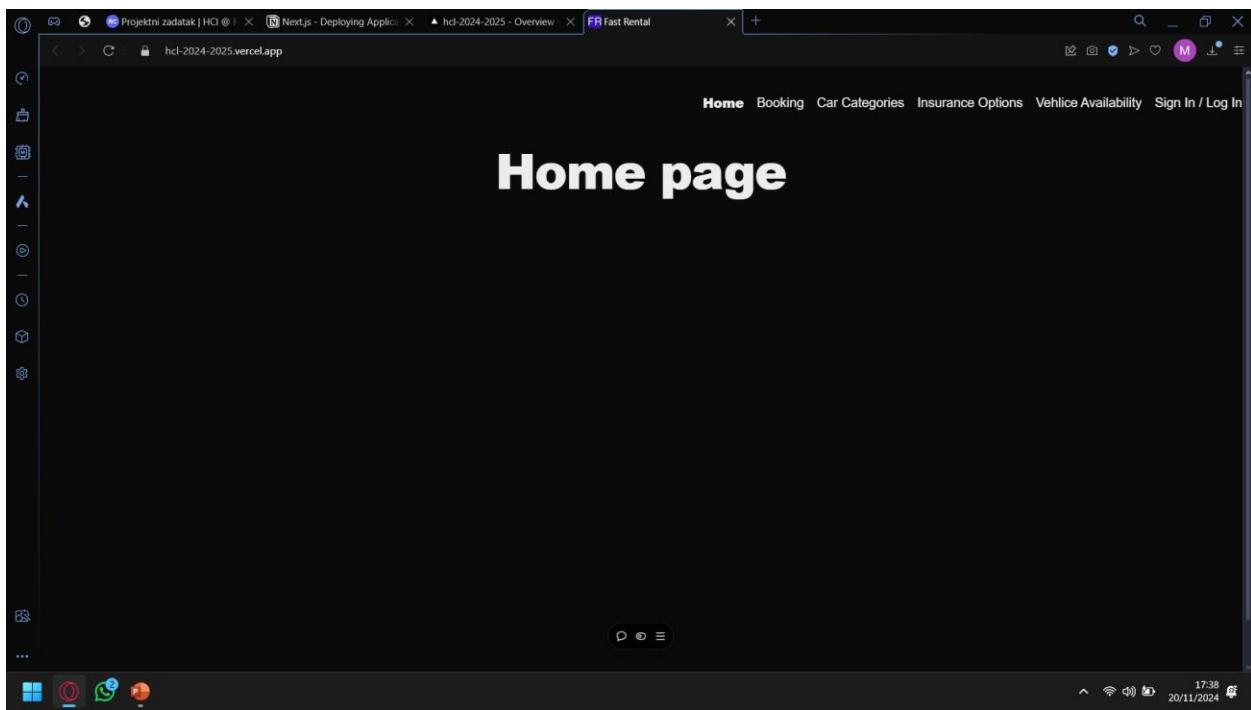
Deploying

One of the requirements of this assignment is also to deploy the application. For that we will use Vercel. It's very simple to do that part since Vercel takes care off basically everything. First you give him permission to use your git repo and then show him Next.js folder in your repo. And that's it, Vercel has deployed your page.

If you want to visit our page Fast Rental, the link is [here](#).

Final Product, so far...

Here is image that shows how our page is looking thus far:



Let's hope it will look much better than this, but for now it will do, after exams we will put a little bit more work into it.

Conclusion

In this short report, we covered what we have done to make the user experience better because no one will wait more than a few seconds to access certain features of your page. The next task is creating low and high fidelity prototypes, in which we will do our best to create a good-looking and user-friendly web application.

Low/High-fidelity prototype

Introduction

In this report, we will explain how we made this task. Here is the table of contents for easier navigation.

[Introduction](#)

[Motivation](#)

[Few technical terms](#)

[Low-Fidelity Prototypes](#)

[High-Fidelity Prototypes](#)

[Key Differences](#)

[Creating Process of High-fidelity Prototype](#)

[Making Color Palette](#)

[Creating templates](#)

[HFP implementation](#)

[Conclusion](#)

Motivation

Low/High-fidelity prototype is a start to every web application. We want users to spend the least amount of time browsing for things they think are important so proper design is a must. Our priority is to make our web application functional and easy to use and design, for now, is second. Of course, we will do our best to make it appealing to our users even though we are not designers 😊

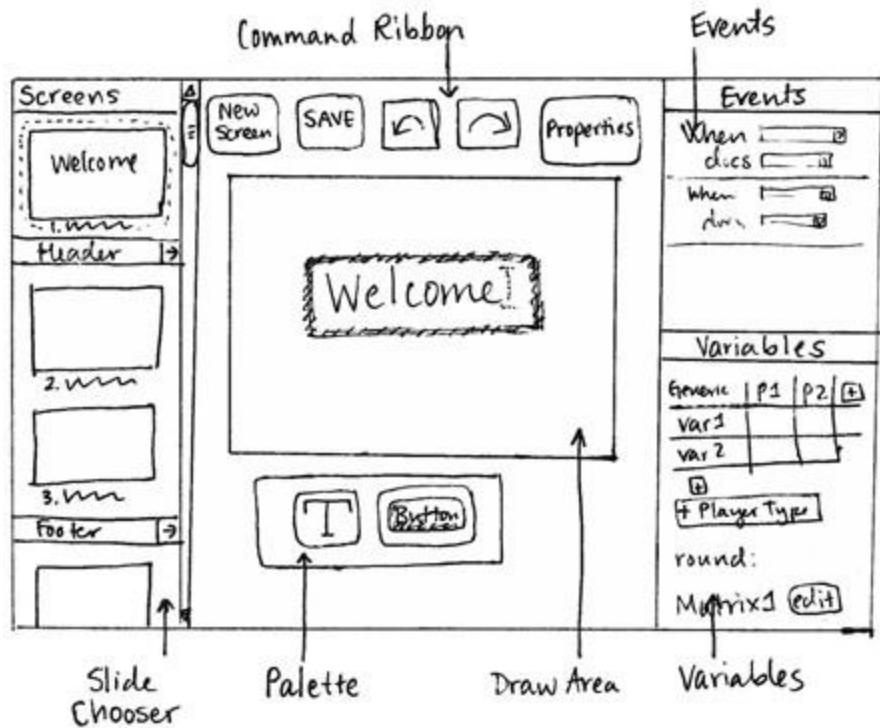
Few technical terms

We used ChatGPT for couple of definitions:

Low-Fidelity Prototypes

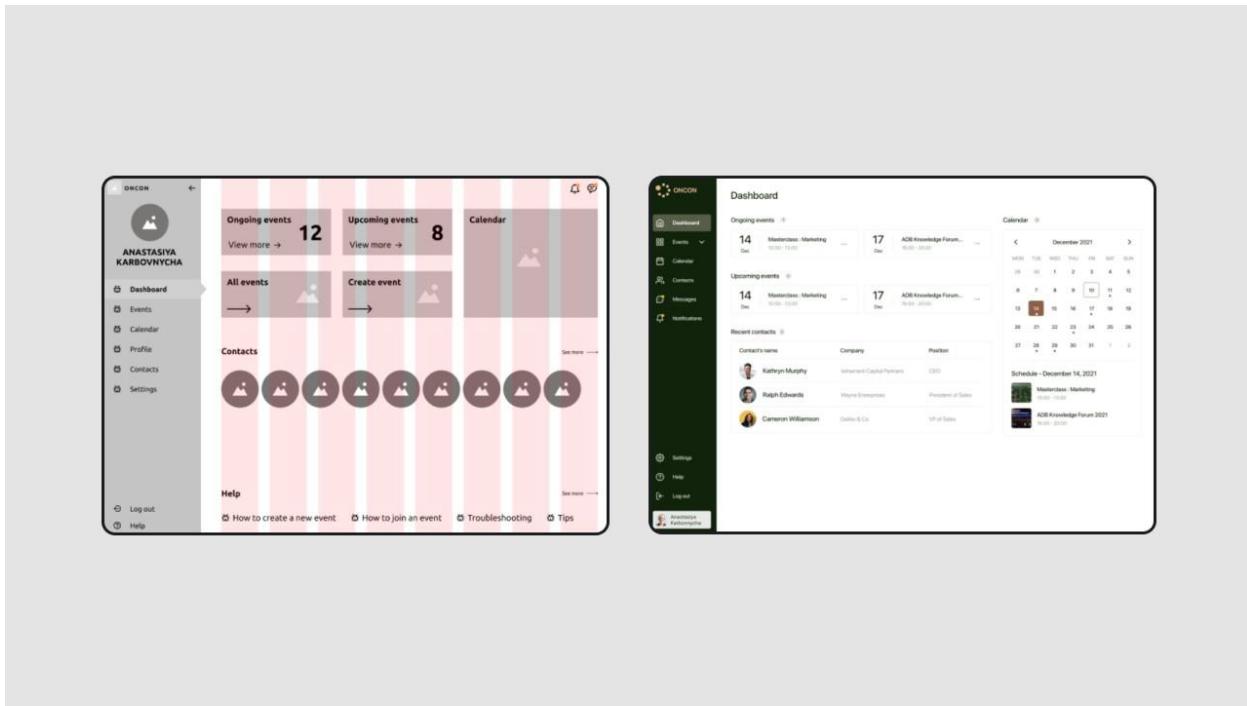
- **Definition:** Simplistic, rough representations of a product or system that focus on the core functionality or layout without detailed design elements.
- **Purpose:** To quickly conceptualize ideas, test functionality, and gather feedback early in the development process.

- Example of one LF Prototype



High-Fidelity Prototypes

- **Definition:** Detailed and polished representations of the final product, closely mimicking its appearance and functionality.
- **Purpose:** To test advanced features, gather detailed feedback, and present a near-final version to stakeholders.
- Example of one HF Prototype



Key Differences

Aspect	Low-Fidelity	High-Fidelity
Detail Level	Basic	Detailed
Interactivity	Minimal	High
Cost and Time	Low	High
Purpose	Conceptualization	Finalization
Tools Used	Paper, Whiteboards	Design software, Code

In this table we can see why and for what we use both Low and High fidelity Prototypes.

Creating Process of High-fidelity Prototype

We, optimistically, decided to go straight to High-fidelity Prototype (HFP from now on) to see how are we gonna implement our web application. When you create HFP in the beginning, during the implementation you do not have to worry about colors, placement, margins and all of that because all of that is available to you. We can divide our “creation” process into 3 main phases:

1. Making color palette

2. Creating templates
3. HFP implementation

We will go through all of them.

Making Color Palette

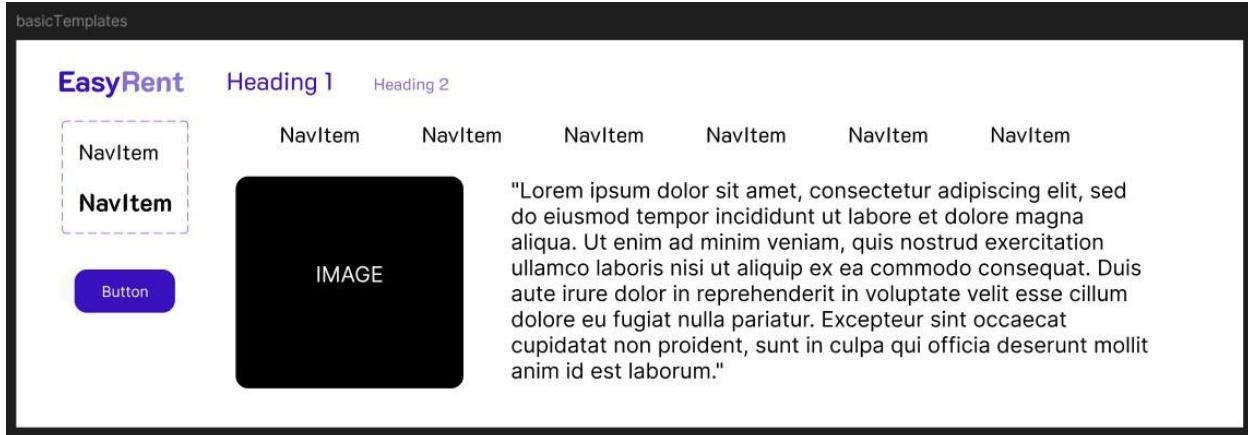
We decided to go with easiest one first. Color palette is important because you will use it all the time and creating one in the beginning is important and it gives us the chance to get familiarize again with Figma.

For our main color we decided to go with a bit lighter blue color. Then we made light and strong variants of that color for text, buttons and so on. You always need black and white but we made white bit darker and black bit whiter so that they are more appealing to your eyes. By mixing those 2 we got grey which we didn't use at the end and we made bit lighter grey for our background. In the end, this is how our palette is looking:



Creating templates

With the easy part out of the way, we spent next 3 days (embarrassingly) creating simple templates for our page. We started with the logo of our web application, Easy_Rent. We combined main and light main color to make simple logo. Next, we created Heading 1, 2 and Paragraph templates as well as button (it is only one for now) and navigation bar utilizing components (active and normal). This is what we came up with:



HFP implementation

And now for the hard part. We decided to follow CRAP as best as possible. If you need a reminder, CRAP stands for:

C - Contrast

R - Repetition

A - Alignment

P - Proximity

How we managed that, you will be the judge of that.

EasyRent

Home Booking Car Categories Insurance Vehicles Sign In/ Log In

Home Page

Looking For Simple, Easy And Fast rental? Look No Further!

Your journey starts here! At Fast Rental, we make renting a car simple, fast, and affordable. Whether you're a young professional heading to a meeting, a student planning an epic road trip, or a retiree exploring new destinations, we've got the perfect ride for you.

Book a Car



Chose The Car That Suits You The Best!



Browse the Cars

Insure Yourself And The Ones You Love!

At Fast Rental, your peace of mind is our priority. We offer three levels of insurance coverage to match your needs and budget. Drive with confidence - select the insurance plan that works best for you!

- Basic Insurance:** Covers damages to the rental car in case of an accident, with a higher deductible. Perfect for confident drivers on a budget.
- Medium Insurance:** Includes collision damage waiver and theft protection with a moderate deductible. A balanced option for extra security.
- Full Insurance:** Comprehensive coverage with zero deductible, covering accidents, theft, and third-party damages. Ideal for worry-free travel.

Check Insurance

Insure Yourself And The Ones You Love!

At Fast Rental, your peace of mind is our priority. We offer three levels of insurance coverage to match your needs and budget. Drive with confidence - select the insurance plan that works best for you!

- **Basic Insurance:**

Covers damages to the rental car in case of an accident, with a higher deductible. Perfect for confident drivers on a budget.

- **Medium Insurance:**

Includes collision damage waiver and theft protection with a moderate deductible. A balanced option for extra security.

- **Full Insurance:**

Comprehensive coverage with zero deductible, covering accidents, theft, and third-party damages. Ideal for worry-free travel.

[Check Insurance](#)

Still Unsure? Read Some Of The Reviews...



As a busy marketing analyst in Split, Fast Rental was perfect for my weekend escapes. The app was easy to use, with great digital payment options. I found an affordable, reliable car and appreciated their excellent service. Highly recommend for anyone who loves hassle-free, budget-friendly travel!

Linda M.



Fast Rental made my trips to visit family so easy and enjoyable. As a retired teacher, I appreciated their friendly customer service and simple booking process. The car was comfortable and reliable for long drives, and the staff was always helpful. Highly recommend Fast Rental for stress-free, senior-friendly rentals!

Mark H.



Fast Rental is a lifesaver for weekend plans! The app is super easy to use, and their prices actually work for a student budget. Plus, they have great options for younger drivers. Totally recommend if you need a quick, affordable ride!

Chris T.



Still Unsure? Read Some Of The Reviews...



As a busy marketing analyst in Split, Fast Rental was perfect for my weekend escapes. The app was easy to use, with great digital payment options. I found an affordable, reliable car and appreciated their excellent service. Highly recommend for anyone who loves hassle-free, budget-friendly travel!

Linda M.



Fast Rental made my trips to visit family so easy and enjoyable. As a retired teacher, I appreciated their friendly customer service and simple booking process. The car was comfortable and reliable for long drives, and the staff was always helpful. Highly recommend Fast Rental for stress-free, senior-friendly rentals!

Mark H.



Fast Rental is a lifesaver for weekend plans! The app is super easy to use, and their prices actually work for a student budget. Plus, they have great options for younger drivers. Totally recommend if you need a quick, affordable ride!

Chris T.



How To Contact And Find Us!



+385 91 234 567



fbrstilo00@fesb.hr



mjuric05@fesb.hr

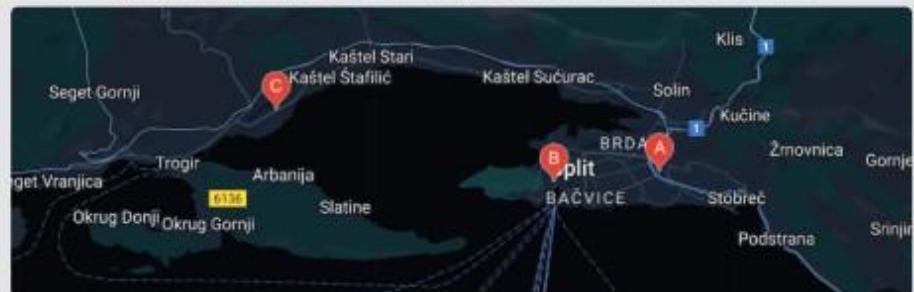


1.) Put Tršćenice 6, Split



2.) Ul. Tomića stine 9, Split

3.) Split Airport, Cesta Dr. Franje Tuđmana 1270, Kaštel Štafilić



EasyRent

© 2024 Brstilo F., Jurić-Pešić M.

We apologize for the low quality of the picture, so on this [link](#) you can see how it looks like exported into PDF.

We followed **Consistency** by making all images having the same:

- Rounding
- Transparency
- Height and width if they are “same level”
- ...

For text it is similar thing:

- Capitalization
- Color and font
- Size of the text
- ...

Repetition has been followed in these things:

- Using same font, colors and font size in text
- Using same property frames around paragraphs

For **Alignment**, we kept everything centered so all the edges on the left of the screen are aligned one to another and same thing is on the right side. Also, all images or text of the same category are all in line.

Proximity is the easy one, all items of same category are all kept at the same place and close to their corresponding title.

Please, let us know if we did this part correctly and if improvements are needed.

Conclusion

In this exercise we made use of Figma to make HFP for our web application. The point of this task is to make user friendly web application that is both functional and appealing to the users. This is NOT final version of our application and during the development we will reconsider our design choices.

Next.js - Dynamic routes, data fetching

Introduction

In this report, we will explain how we made this task. Here is the table of contents for easier navigation.

[Introduction](#)

[Motivation](#)

[Technical Part](#)

[Fixed design](#)

[Additions](#)

[Before vs After](#)

[Before](#)

[After](#)

[Conclusion](#)

Motivation

Not every time, more like 99% of the time, you will have a database (DB) from now on, that you will NEED to have in order to get your data onto a website. Also, when users are signing in or making local changes, we need to store that somewhere. That is why you need a DB.

Technical Part

We used [Car Categories Page](#) for this part of the exercise. We will use this page later on to fetch our custom data of the cars. We will also do sorting, searching and all of the other task requirements. Steps of the implementation:

1. Create fetchData function
2. Parse data
3. Display data
4. Pagination

This is all technical stuff, You can see it all on [Github](#). Here is the image of fetched data and pagination:

The screenshot shows a dark-themed web application titled "EasyRent". At the top, there are navigation links for "Home", "Booking", "Car Categories" (which is the active tab), and "Insurance". A "Sign In / Log In" button is also present. Below the navigation, there are four categories: "Compact", "Medium", "Large", and "Vans". The main content area is titled "Car Categories" and features a heading "Fetched Posts:". Below this, there are three card-like boxes containing text snippets:

- 1. sunt aut facere repellat provident occaecati excepturi optio reprehenderit**
quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto
- 2. qui est esse**
est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla
- 3. ea molestias quasi exercitationem repellat qui ipsa sit aut**
et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

The screenshot shows the same "EasyRent" application. The "Car Categories" section displays four card-like boxes with text snippets:

- 17. fugit voluptas sed molestias voluptatem provident**
eos voluptas et aut odit natus earum aspernatur fuga molestiae ullam deserunt ratione qui eos qui nihil ratione nemo velit ut aut id quo
- 18. voluptate et itaque vero tempora molestiae**
eveniet quo quis laborum totam consequatur non dolor ut et est repudiandae est voluptatem vel debitis et magnam
- 19. adipisci placeat illum aut reiciendis qui**
illum quis cupiditate provident sit magnam ea sed aut omnis veniam maiores ullam consequatur atque adipisci quo iste expedita sit quos voluptas
- 20. doloribus ad provident suscipit at**
qui consequuntur ducimus possimus quisquam amet similique suscipit porro ipsum amet eos veritatis officiis exercitationem vel fugit aut necessitatibus totam omnis rerum consequatur expedita quidem cumque explicabo

At the bottom of the page, there are "Previous" and "Next" buttons.

If You want to see production version here is the link:

<https://hcl-2024-2025.vercel.app/>

Or if You want to see just the pagination, here is a direct link:

Easy Rent

Your car is but a few clicks away

 https://hcl-2024-2025.vercel.app/car_categories

Fixed design

On our last meeting, we discovered some issues. We used this time to fix those issues. Some of them were:

- Title too big and distracting
- Navigation spacing and number of navitems
- Removed distracting Home Page text
- Call-to-action button made more important
- Removed unnecessary text from insurance tag
- Reviews design fixed
- Information area redesigned

Additions

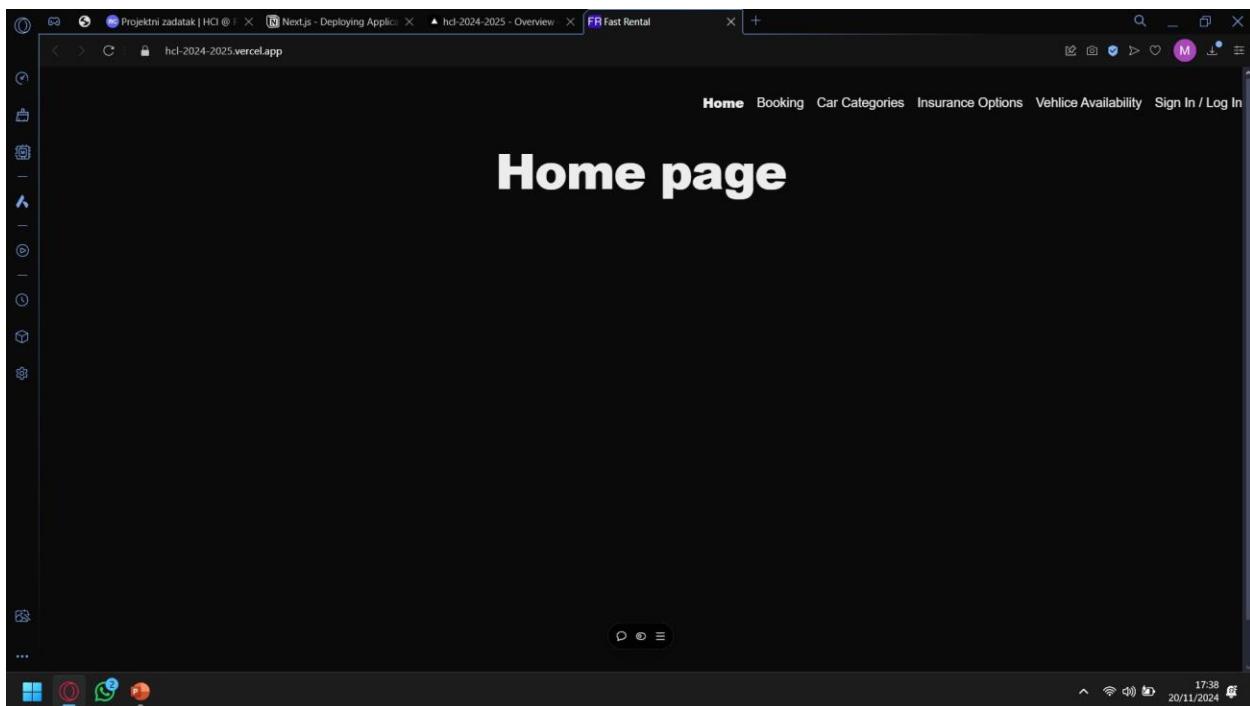
We also made some additions as well, for example we made:

- Footer as separate component
- Car categories as data fetching site
- Booking page
- ...

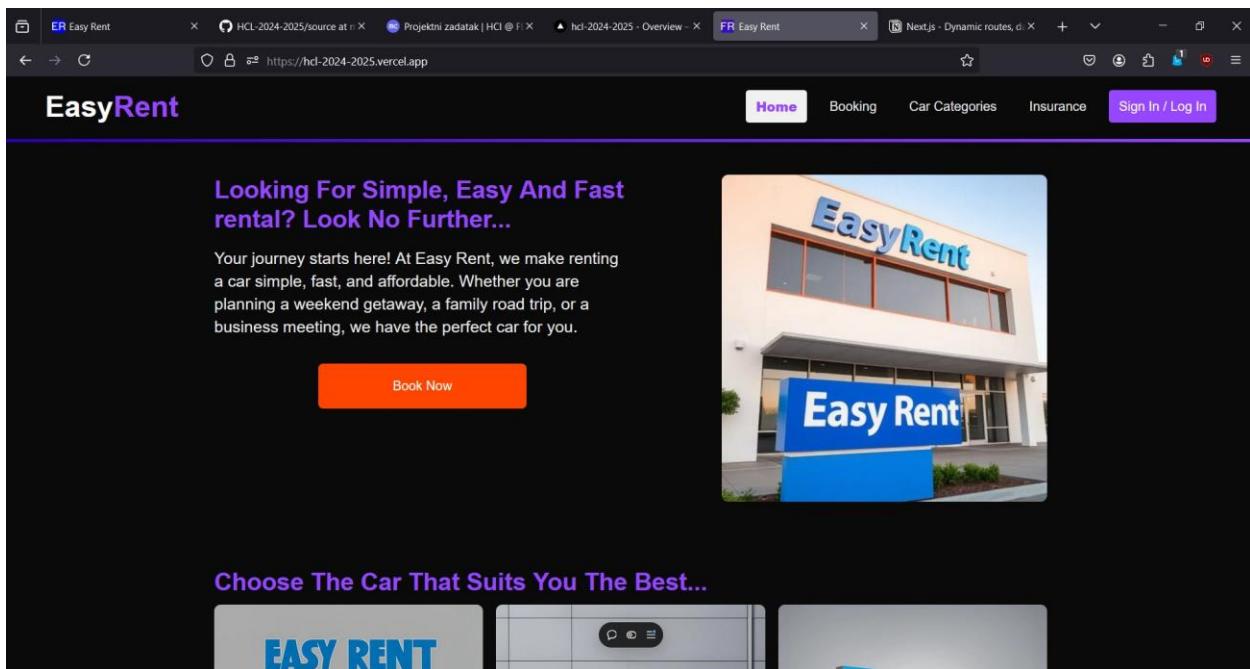
Before vs After

After we made all of those modifications, we can compare pages from previous task and this task.

Before



After



Conclusion

We made a lot of progress, but it is only the beginning of the successful web application. In the future tasks, we will try to improve on our design and our skills.

Full Responsive Page Coding

Introduction

In this report, we will explain how we made this task. Here is the table of contents for easier navigation.

[Introduction](#)

[Motivation](#)

[Technical Part...](#)

[Key Features of Tailwind CSS](#)

[Result](#)

[Desktop](#)

[Mobile](#)

Motivation

Creating a fully responsive page is essential in today's digital landscape, where users access websites on a variety of devices, including smartphones, tablets, laptops, and desktops. The goal of responsiveness is not just about aesthetics, it is about delivering a seamless and optimized user experience across all screen sizes and devices.

Technical Part...

For CSS and responsive design, we used Tailwind CSS in order to achieve responsive design. It is a utility-first framework that simplifies the process of creating responsive layouts. Tailwind CSS allows for rapid development and provides a rich set of pre-defined classes, enabling us to implement responsive designs without writing custom CSS for every element.

Key Features of Tailwind CSS

The framework is widely used for its scalability, performance, and ease of use. Once you get used to it it is really fun to use too. Another thing is that you don't have to make global CSS files, you just write in components, how fun is that?

Some of key features are (using ChatGpt)

- **Utility-First Classes:** Provides a vast library of pre-defined utility classes for rapid styling without writing custom CSS.
- **Responsive Design Utilities:** Built-in classes like `sm:`, `md:`, `lg:`, and `xl:` enable easy implementation of responsive designs.
- **Customization:** Offers deep customization through the `tailwind.config.js` file to adjust themes, colors, and spacing.
- **Flexbox and Grid Support:** Simplifies the creation of complex layouts with intuitive flexbox and grid utilities.
- **Performance Optimization:** Includes a built-in `purge` feature to remove unused CSS, reducing file size for production builds.

Result

If you need a reminder, here is how our landing page is supposed to look when we first made it:

Desktop

EasyRent

- Home
- Booking
- Car Categories
- Insurance
- Vehicles
- [Sign In/ Log In](#)

Home Page

Looking For Simple, Easy And Fast rental? Look No Further!

Your journey starts here! At Fast Rental, we make renting a car simple, fast, and affordable. Whether you're a young professional heading to a meeting, a student planning an epic road trip, or a retiree exploring new destinations, we've got the perfect ride for you.

[Book a Car](#)



Chose The Car That Suits You The Best!



[Browse the Cars](#)

Insure Yourself And The Ones You Love!

At Fast Rental, your peace of mind is our priority. We offer three levels of insurance coverage to match your needs and budget. Drive with confidence - select the insurance plan that works best for you!

- **Basic Insurance:** Covers damages to the rental car in case of an accident, with a higher deductible. Perfect for confident drivers on a budget.
- **Medium Insurance:** Includes collision damage waiver and theft protection with a moderate deductible. A balanced option for extra security.
- **Full Insurance:** Comprehensive coverage with zero deductible, covering accidents, theft, and third-party damages. Ideal for worry-free travel.

[Check Insurance](#)

Mobile

EasyRent



Looking For Simple, Easy And Fast rental? Look No Further...

Your journey starts here! At Easy Rent, we make renting a car simple, fast, and affordable. Whether you are planning a weekend getaway, a family road trip, or a business meeting, we have the perfect car for you.

[Book Now](#)



Choose The Car That Suits You The Best...

EASY RENT



Insure Yourself And The Ones You Love...

Basic Insurance

Covers damages to the rental car in case of an accident, with a higher deductible. Perfect for confident drivers on a budget.

Medium Insurance

Includes collision damage waiver and theft protection with a moderate deductible. A balanced option for extra security.

Full Insurance

Comprehensive coverage with zero deductible, covering accidents, theft, and third-party damages. Ideal for worry-free travel.

[Check Insurance](#)

How To Contact And Find Us...



+385 91 234 567



fbrstilo00@fesb.hr
mjuric05@fesb.hr



- A Put Trščenice 6, Split
- B Ul. Tomića stine 9, Split
- C Split Airport, Cesta Dr. Franje Tuđmana 1270



EasyRent

© Brstilo, Jurić-Pešić

We made some changes to mobile version brought our work to make it look similar to our last version that you can find here: [Easy Rent](#).

Note that ALL pages are coded to fit all screens.

FINAL REPORT

Introduction

This document marks the conclusion of our semester-long journey in building the Easy Rent web application. What started as a simple prototype in Figma gradually transformed into a fully coded, deployed, and responsive web application hosted on Vercel.

Throughout this process, we combined theory with practice: learning how to apply UI/UX principles, using modern frameworks such as Next.js with Tailwind CSS, and following structured HCI methodology that emphasized user personas, card sorting, information architecture, prototyping, and iterative design.

The purpose of this final report is to reflect on the progress we made, document the changes from the initial prototype to the final deployed version, and evaluate the overall results, including a performance test with Google's Speed Insight.

Evolution of the Project

1. Figma Essentials

- We began with simple frames, navbars, and hero sections. The goal at this stage was to learn the tools and create a flexible design base using assets and instances for reusable components.

2. User Personas & Information Architecture

- Personas such as *Sarah Lee (Young Professional)*, *Robert Carter (Retired Senior)*, and *Javier Martinez (College Student)* helped us align our design with actual user needs.
- Card sorting and sitemap creation shaped the content hierarchy, ensuring the website remained intuitive and user-friendly.

3. Next.js Deployment

- Transitioning from static prototypes, we implemented routing, layouts, and deployed our first functional version on Vercel.
- This milestone taught us about continuous deployment and how small iterations can quickly reach production.

4. Low & High Fidelity Prototypes

- Using Figma again, we refined our visual style with a custom color palette, consistent typography, and reusable UI templates.
- The principle of CRAP (Contrast, Repetition, Alignment, Proximity) guided us in making the design professional and consistent.

5. Dynamic Routes & Data Fetching

- We integrated Car Categories with dynamic routing and implemented pagination to simulate real-world data handling.
- This step marked the transition from a static project to a more interactive and

scalable solution.

6. Full Responsive Coding

- Using Tailwind CSS, we ensured that every page works seamlessly across mobile, tablet, and desktop devices.
- Responsiveness was not only a design choice but also a performance requirement, as modern users expect seamless browsing regardless of device.

Before vs. After: What Changed?

From the initial design to the final product, the Easy Rent application underwent major improvements:

- **Navigation Bar**
Originally cluttered and repetitive ("home home home"), it was redesigned into a clean, intuitive structure with meaningful labels.
- **Hero Section & Call-to-Action**
Early drafts contained oversized titles and placeholder text. In the final version, the hero is balanced with clear CTAs and a more professional look.
- **Insurance Plans**
The placeholder text was replaced with meaningful Basic, Medium, and Full coverage cards that improve readability and visual appeal.
- **Customer Reviews**
Redesigned to appear cleaner and less distracting, with emphasis on testimonials rather than filler text.
- **Footer**
Updated to include branding (EasyRent) aligned left and copyright (© 2024 Brstilo F., Jurić-Pešić M.) centered.
- **Responsiveness**
The entire application now adapts smoothly across devices, unlike the initial versions which were designed mainly for desktop.

Page & Design Changes Over Time

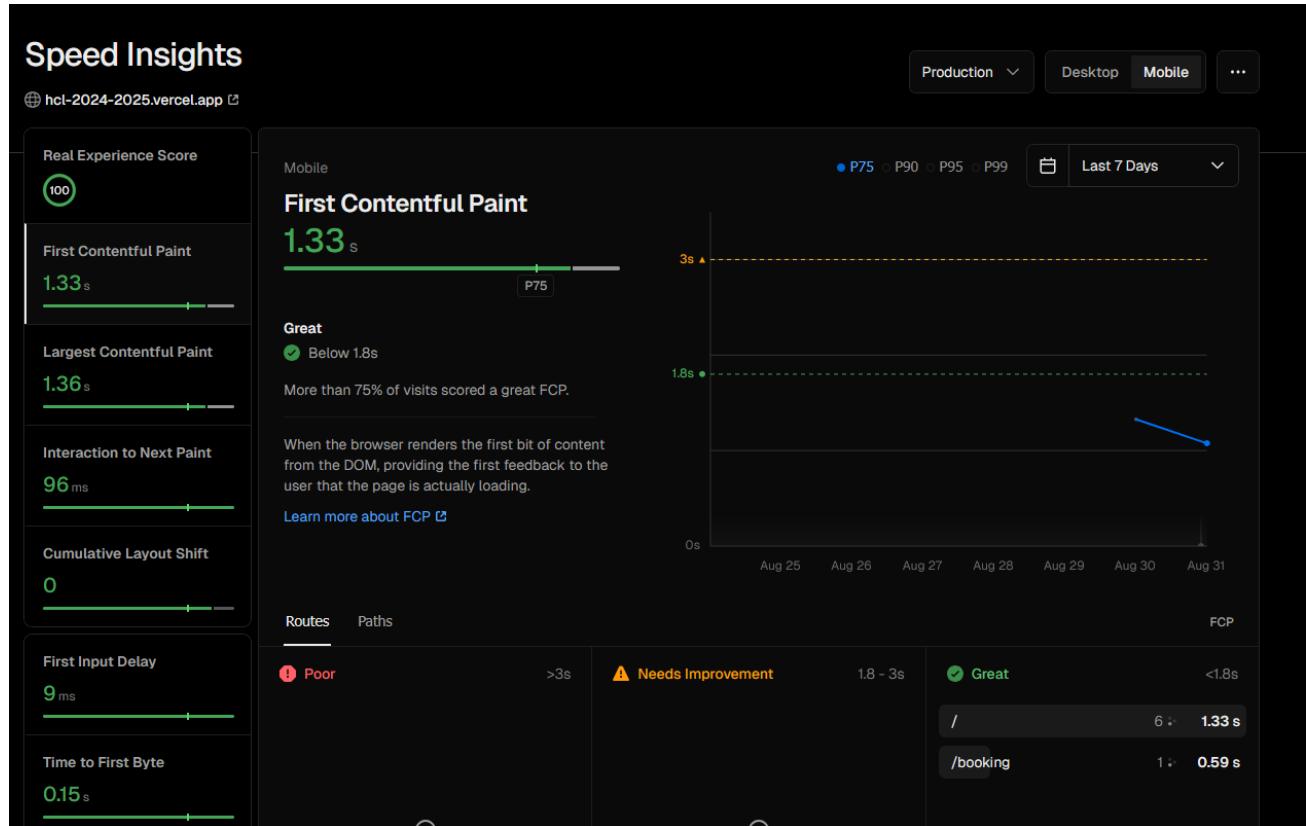
- Initial Figma phase: ~2 static prototype pages (hero + navbar)
- Personas & IA phase: sitemap planned with 6 core sections
- Next.js first deployment: 6 navigation routes created
- Dynamic Routes phase: added Car Categories, Booking, Insurance
- Final version: fully responsive multi-page application with over 10 functional pages/components redesigned from scratch.

In total, we can conclude that more than 8 pages/components were redesigned and

refined between the initial and final stages.

Performance Results (Vercel Speed Insight)

To evaluate the technical quality of the final product, we tested the deployed version using Vercel's PageSpeed Insight. Below is the final screenshot of performance results:



This test confirms that the Easy Rent application is both visually optimized and technically performant, providing users with a smooth browsing experience.

Conclusion

The Easy Rent project demonstrates how an idea can evolve from a conceptual Figma wireframe into a fully deployed and responsive web application.

Key takeaways:

- **Prototyping first saves time:** By validating designs early, we avoided costly redesigns during coding.
- **User personas drive usability:** Tailoring the site for different demographics improved navigation clarity.
- **Iteration is essential:** Our “Before vs After” comparisons highlight how much refinement happened in each step.
- **Modern frameworks accelerate growth:** Next.js and Tailwind CSS allowed rapid prototyping, responsive design, and easy deployment.

- **Performance matters:** By optimizing structure and responsiveness, we ensured a positive experience across devices.

In conclusion, our project journey covered not only technical implementation but also design thinking, user empathy, and performance optimization. The result is a complete Rent a Car application prototype that is both user-friendly and ready for further development.