



COMP 200

Summer Practice Report

Enes Ertuğrul
042101074

16.07.2025-20.08.2024, 24 workdays
Submitted: Report Submission Date

MEF University

Computer Engineering Program
Internship Report

Executive Summary

In the course of my internship at iCredible Technologies, I obtained numerous experience in software development methodologies. I was able to work on projects with technologies I've never even touched and get a feel for how things are done in a team environment. This experience gave me more than just technical skills, but what it means to work in business and with a team.

My internship lasted five weeks. In this period I attempted to comprehend team members and derived from their standpoints. I also did heavy work for the project I had been assigned. I developed a company's CSMS as part of the project. The process was intense and incredibly cool.

Week 1 I spent the whole week researching what technologies I would have to use. I had technologies that I never tried before, like .NET, Docker, Postman, and DBeaver. I presented these tools briefly in front of my mentor, and I remembered by doing that I was also reviewing those tools.

I started the 2nd week working exclusively on the main design of my project. I also wireframed the pages with Figma, made my database schema and set up all API endpoints. This step was to get the nuts and bolts of this project in place.

After the third week I started with front-end development. I faced with different problems on every step of this way, but my mentor always help me to solve it. This really illustrated the value of problem solving ability, knowing when to ask the right questions, and the importance of clear internal communication."

In the fourth week, I started developing the backend. I used .NET for the backend and PostgreSQL for the database. Since this was my first time doing both, I struggled at times. Working with my mentor to solve problems not only shortened my learning curve but also gave me real-world experience managing a professional work environment with a collaborator.

I integrated the frontend and backend modules and completed my project last week. I learned a lot about integrating the frontend and backend, and I witnessed how the created file structures and DTOs facilitated communication. This process helped me understand the importance of combining various components to create a complete system.

This internship allowed me to experience implementing technologies I had never used before and significantly contributed to my technical development. How an application should be planned, the frontend, backend, and database structure, and doing everything myself, taught me a lot about the entire project.

This gave me a better understanding of the interconnectedness of the technologies and how they communicate. Also, observing office life, seeing how work gets done and how people interact as a group, was invaluable. I saw firsthand the importance of teamwork, how people solve problems, how people's thought processes work, and how people become productive.

Contents

Executive Summary.....	1
Contents.....	2
1. Company and Sector Overview.....	3
1.1 Overview of the Company and Sector	3
1.2 Organization of the Company	4
1.3 Production Service System	5
1.4 Professional and Ethical Responsibilities of Engineers.....	5
2. Summer Practice Description	6
2.1 Activity Analysis	6
2.2 Conclusion	7
2.3 Impact	7
2.4 Self-Directed Learning	7
3. Appendix: Daily Activity Table	7

1. Company and Sector Overview



Figure 1: The logo of iCredible Technologies

Address: Barbaros, Sebboy Street No:4/1 Floor:12 Office:56 34634 Atasehir/Istanbul, TR

1.1 Overview of the Company and Sector

iCredible Technologies, Inc. is a Pacific-based Secure and innovative blockchain solution provider. Positioned within fintech and cybersecurity, iCredible provides a diverse range of products and services to address the increasing global need for reliable protection and management of digital assets.

One of iCredible's primary areas of expertise is regulatory compliant crypto asset custody solutions serving institutional and retail investors. iCredible also builds transparent and reliable cryptocurrency exchanges for streamlined digital asset trading.

iCredible also provides custody services for scalable and resilient platforms used for safe storage of cryptocurrencies and tokens. iCredible's tokenization and NFT marketplace innovations allow physical and digital assets to be interchangeable on blockchain networks. Advanced digital economy.

To ensure optimized asset management, iCredible Technologies helps investors with applications for asset management with functionalities for effective supervision, performance tracking, and risk mitigation. In addition to financial services, the company has blockchain-based endpoint encryption for data isolation, thus protecting exposed information against loss or unauthorized access.

Also, iCredible Technologies has a fragmented custody system in which stored assets are divided along various dimensions or axes for the purpose of risk reduction and increasing the system's robustness. These custodial systems are layered with diverse custody systems, which secured financial systems.

iCredible Technologies' integrated custody functionalities continue to contribute to the adoption of digital assets and advances in blockchain technology, not only locally but also globally.

1.2 Organization of the Company

iCredible Technologies is a technology company focused on developing data solutions designed to protect digital assets. The company designs innovative software and systems that safeguard the information assets of diverse industries, offering proactive measures against cyber threats and fostering secure digital environments.

Through its commitment to combining with advanced security frameworks, iCredible Technologies aims to provide organizations with reliable, scalable, and forward-looking tools that enhance resilience in the face of evolving digital challenges.

The organization is structured into several departments, including:

Front-End Development: Responsible for designing and implementing user-friendly, responsive interfaces to ensure seamless customer interaction with security platforms.

Back-End Development: Focuses on building robust server-side architectures, API services, and database management systems critical to secure data processing and integration.

Software Engineering: Integrates AI algorithms and cybersecurity functions into compatible, scalable systems while maintaining application performance and security compliance.

Business Strategy and Marketing: Manages market positioning, customer engagement, and growth strategies to expand iCredible's impact in the cybersecurity industry.

Mustafa El Aliwat - Co-Founder & CEO
Muhsin Artıranlar - Lead Software Architect
Cihan S. - Tech Lead & Project Manager
Murat Kececi - Senior Software Developer
Volkan Y - Senior Software Developer
Mete Çoşkun - Account Manager
Nur Betül İyican - Software Developer

1.3 Production Service/System

The company focuses on delivering advanced data and asset security. Integrating blockchain with preventative cybersecurity technologies, they offer trust and effectiveness in protecting clients' digital assets and making them adaptive to dynamic cyber threats.

The company's service system is divided into three main stages:

1. Digital Asset Custody and Exchange: The firm offers compliant custody solutions alongside SPK cryptocurrency exchange facilities, permitting clients to trade virtual currencies and secure digital assets.

2. Tokenization and Portfolio Management: iCredible Technologies fosters digital economy engagement through efficient investment services and constructs NFT marketplaces, as well as developing portfolio management applications.

3. Blockchain Security and Risk Reduction: Embedded virtual encryption and fragmented custody systems offer endpoints security and comfort regarding digital assets management.

1.4 Professional and Ethical Responsibilities of Engineers

During my internship as a Software Engineer intern at iCredible Technologies, even though I did not perform any active duties, I had a great chance to monitor the professional and ethical duties concerning the engineering projects. The primary professional and ethical duties that engineers must take in the process of software development consist of:

- Design and uphold the safety of blockchain-based systems while guaranteeing digital assets custody and management platforms operate continuously, faultlessly, and reliably.
- Guard client data and digital assets while conforming to global data security frameworks and statutory requirements, including the GDPR.
- Ensure openness of system design and functionality, providing thorough and intuitive explanations of blockchain, tokenization, and encryption.
- Equitably distribute digital services by creating inclusive designs to ensure no user is excluded.
- Enhance security and cyber threat detection and prevention, and implement risk-mitigating measures, such as split custody systems, by using sensitive data to a necessary degree.

These primary ethical values that must be observed at all times in the engineering profession and even at the cybersecurity companies like iCredible Technologies. The values and the ethical observance in my internship were the key inbuilt observance of professional careers

2. Summer Practice Description

2.1 Activity Analysis

During my software engineer internship at iCredible Technologies, I was involved with different stages of the project development lifecycle. This activity was designed to give me insight into various software development processes.

1. Week 1: Technology Research and Presentations

Initially, I studied the technologies that I was going to work with for the project. Some of the technologies were unfamiliar and I had to learn more about them to understand the basics. I consolidated my findings and presented them to my mentor, who then provided me with feedback. This helped to build my understanding of the various technological facets of the project.

2. Week 2: Project Planning and Design

During the second week, I was responsible for creating the project frontend design using Figma. I also designed the initial framework of the database, and constructed some of the backend APIs. This was crucial for me to comprehend the project's architecture and prepare to build the next layers confidently.

3. Week 3: Frontend Implementation with React and NextJs

In the third week, I finished the frontend coding process with React and NextJS. I focused on creating a user interface that was usable and maintainable, implementing modern and type-safe technologies. Interface development provided a unique technical learning experience this semester.

4. Week 4: Backend Development

In week four, I transitioned to coding the backend. I developed API integration, along with the necessary connections to the system's databases. This was essential to building the system's foundational components relating to the backend data management and the core business logic.

5. Week 5: Integration and Testing

In week five, I merged the code from both the frontend and backend to maintain integrity of the whole system. This allowed the different segments of the system to function in unison, upon which the necessary functionality tests were performed. This was the last integration phase required for the system to work in full functionality.

2.2 Conclusions

During my internship, I developed a comprehensive project, handling both the front-end and back-end design and coding myself. This process enhanced my understanding of project management, software architecture, and system integration. I also spoke with employees at the company, sharing insights into their work and experiences, and trying to learn from them. Working alongside cybersecurity and blockchain experts increased my awareness of critical industry considerations and challenges, allowing me to focus on core issues.

2.3 Impact

At my internship, I independently built an entire project from the ground up, which included the frontend, backend, and database, and this was my first exposure to full-stack development. Although I didn't work on the company's actual projects, I was able to observe the work of the cybersecurity team. This experience gave me insight into the intricacies of the technology and the requirements of the field. This was the first time I had the opportunity to legally practice my software development skills, and it also gave me experience in the blockchain and cybersecurity industries. This helped to shape my practical skills and gave me a solid idea for my future career.

2.4 Self-Directed Learning

In my internship, I encountered .NET, Docker, Postman, and DBeaver for the first time while working on project development. I made sure to allocate additional time for my personal project to gain a deeper understanding of these tools. I explored the documentation, leveraged the internet, carried out mini-projects, and maintained communication with my mentor. These spaced out efforts helped me understand and utilize the technologies to the project.

Appendix 1: Daily Activity Tables

Day 1:

When I 1st started at my new company, the team all went out of their way to greet me and meet everyone. My supervisor gave me a nice run-down of what an internship at this organisation looks like and the responsibilities for my role. I got a sense of the company culture, their values and how various departments work together to meet common goals. That orientation set a really good tone for how I could come into the workplace and gave me confidence about what to expect in the coming weeks.

Date	Supervisor's Name	Signature
16.07.2025	Betül Nur İyican	

Day 2:

My mentor and I went through the internship papers on day two. We also went through the duties, expectations and milestones, I must achieve throughout my internship. This procedure cleared up all formal and procedural issues, and my advisor also helped me answer some of the questions I had about how project work is carried out. The co-review process worked to make sure I was well aware of my obligations and development path during the internship.

Date	Supervisor's Name	Signature
17.07.2025	Betül Nur İyican	

Day 3:

And here my teacher gave me about 15 things that I had to go research and present. All day getting together this presentations, looking information for I can see the technology and project needs. Later on, I presented what I found out to my teacher. This became a good opportunity for me to improve communication with more experienced coworkers and gain confidence in speaking technical terms.

Date	Supervisor's Name	Signature
18.07.2025	Betül Nur İyican	

Day 4:

The fourth day was just for me to do some research into the tech I would be working with. I compared a number of development environments and frameworks, including what type they where, pros/cons, and typical purpose. I pick up a bunch of stuff about React. This feasibility study enabled me to, and facilitated practical application.

Date	Supervisor's Name	Signature
21.07.2025	Betül Nur İyican	

Day 5:

Then I learned even more about things, i.e. read documentation, etc, from the day before. I also shared my learnings with my mentor to confirm my decisions and get guidance on idioms. When i said my idea I had a clearer concept on the mind. This form of high level preliminary design research is crucial to establish a good base of knowledge, before hardened development began.

Date	Supervisor's Name	Signature
22.07.2025	Betül Nur İyican	

Day 6:

I spent the day trying to clear up the project idea. I started by having an in-depth discussion with my mentor to establish the basic functionality of the application, what kind of user roles would be involved and how it would flow. By talking through use cases, I developed a click-through user flow for users signing in to the system and device management. I did also taken into account when making the decision, functional and non-functional requirements (non-functional ones such as performance and usability). I also outlined the preliminary system architecture and determined which components would be focused on initially. For instance, I put together a simple architecture diagram that outlined exactly how the front-end, the back-end, database and API would communicate. This was key for accelerating the development phases and working in a more organized and predictable manner.

Date	Supervisor's Name	Signature
23.07.2025	Betül Nur İyican	

Day 7:

Then I went onto designing the front end. First step was I searched for a pre-made Figma design and adjusted it to fit my project. I took care of UI, the look and feel of main screens such as homepage, device management (room), room management itself and profile page. Throughout the process, I took into account UI/UX principles such as accessibility-friendly color contrasts, sensible icon placement, and friendly transitions. I would also go over the designs with my mentor for input. He liked the design in general and that I was methodical.

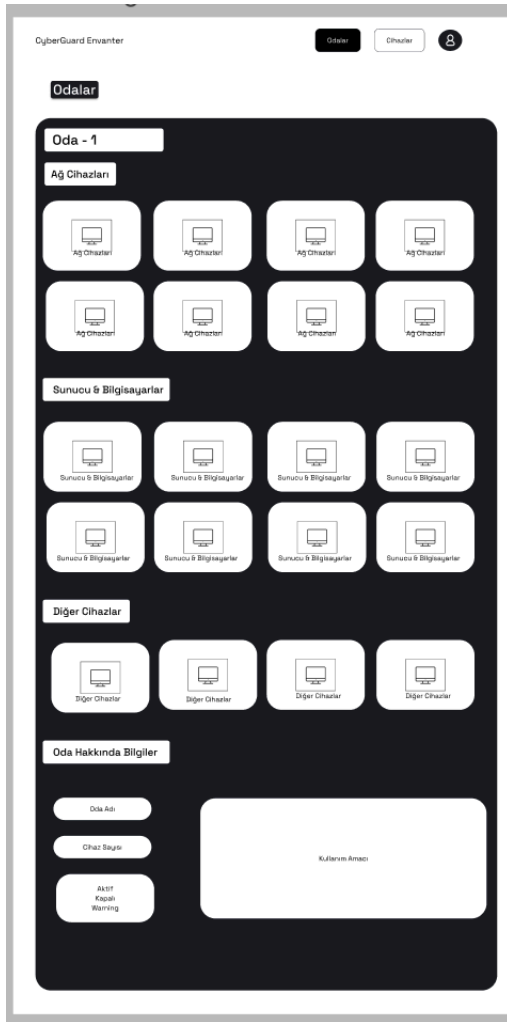


Figure 2: Figma design

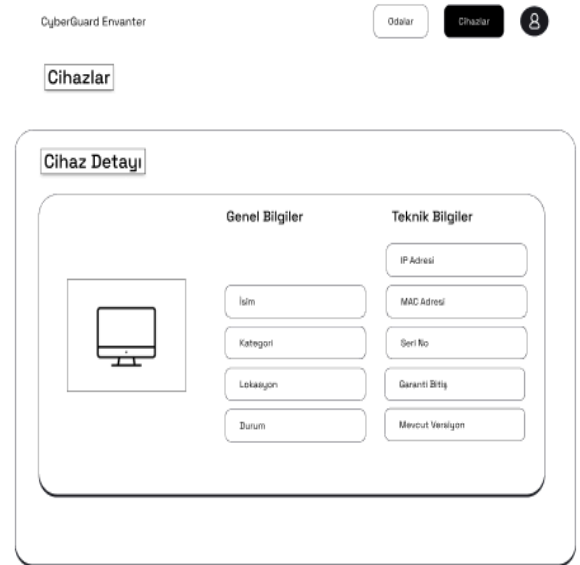


Figure 3: Figma design

Date	Supervisor's Name	Signature
24.07.2025	Betül Nur İyican	

Day 8:

I type of continued from yesterday, working on the design of the admin panel. I have used a similar visual language to remain consistent with the user side designs, but from admin side there are extra permissions and different functions. For instance, I developed separate module for admin operations like Add Device, Delete Device, Update Device and so on. This made it possible for the admin interface to provide different functionality from the user, as onboarding logic closely follows the RBAC. This provided the admin with a UI that could handle many devices in an easily navigable way. By the end of the day I managed to produce a full Figma prototype, both for the user and admin dashboard. This prototype was later turned into a design specification that the i could refer to during the development.

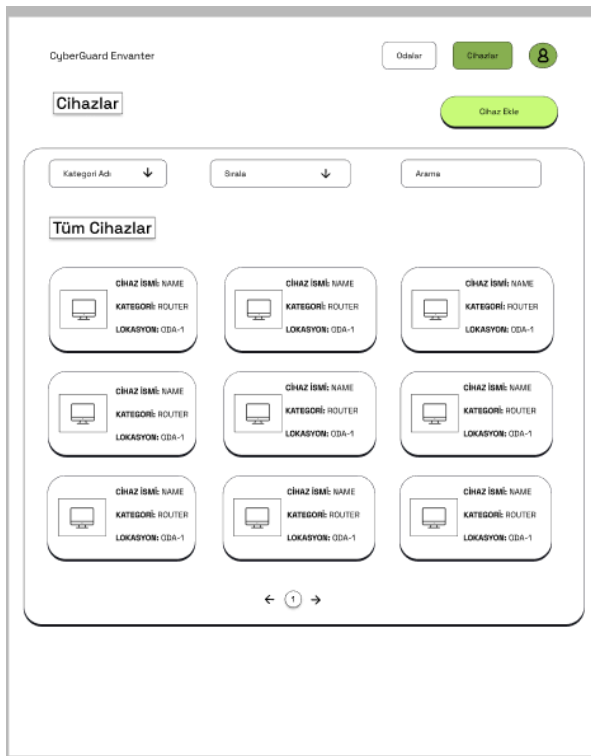


Figure 4: Admin panel design



Figure 5: Admin panel design

Date	Supervisor's Name	Signature
25.07.2025	Betül Nur İyican	

Day 9:

I spent the day on designing databases. I made an ERD with PgSQL to visualize some of the relationships between tables. I had defined simple tables like user details, device information, rooms and 'users and room' relationship. After that I had to set fields for the tables (like user table: ID, name, email and password hash | device table: ID, device name, status information and room id). I used principles of normalization to guide data structure decisions. This in turn worked towards eliminating data redundancy and aiming to make it a high performance database that is scalable. I also worked on the database so as to fit the UI requirements and I considered where what should be listed, which data we filter on frontend etc. This approach provided a data flow base for the app.

Date	Supervisor's Name	Signature
28.07.2025	Betül Nur İyican	

Day 10:

I started to design APIs on the backend. The first plan I had was to create two distinct API sets: one for users and another for the administrators. I established the endpoints for these APIs, as well as the HTTP methods with which they would be accessed and then request-response structures for each endpoint. i.e On the user side, I created something like GET /devices to get the list of devices and PUT /devices/:id to update device status. On the admin side, I created functions like POST /devices for adding devices and DELETE /devices/:id for deleting them. I also took care of authentication and authorization while doing so. I had lines on paper for layers of security, like user auth via JWT & additional permission system applied to the admin actions. I also needed to standardize error messages, have consistent responses and API documentation generated using something like Swagger. This means frontend and backend teams could "speak the same language", and series of future possible integration problem was even lessened.

POST	/api/devices	Yeni cihaz ekler	Admin
PUT	/api/devices/{id}	Cihaz bilgilerini günceller	Admin
DELETE	/api/devices/{id}	Cihazı siler	Admin

AUTHENTICATION ENDPOINTS

Method	Endpoint	Açıklama	Yetki
POST	/api/auth/login	Giriş işlemi	Herkes
POST	/api/auth/register	Kayıt işlemi	Herkes

Figure 6: API endpoints

API ENDPOINTS

USER/ADMIN

Method	Endpoint	Açıklama	Yetki
GET	/api/rooms	Odaların listesini getirir	Normal/Admin
GET	/api/rooms/{id}	Belirli odanın detaylarını getirir	Normal/Admin
GET	/api/devices	Tüm cihazların listesini getirir	Normal/Admin
GET	/api/devices/{id}	Belirli cihazın detaylarını getirir	Normal/Admin

ADMIN

Method	Endpoint	Açıklama	Yetki
POST	/api/rooms	Yeni oda ekler	Admin
PUT	/api/rooms/{id}	Oda bilgilerini günceller	Admin
DELETE	/api/rooms/{id}	Odayı siler	Admin

Figure 7: API endpoints

Date	Supervisor's Name	Signature
29.07.2025	Betül Nur İyican	

Day 11:

I spent today managing project file structure. Firstly I sorted out how my file structure would look, creating folders for components, services, assets and configuration. > This was to avoid any future confusion in both front end and backend. De-coupling the code in a modular fashion not only speeded up development, but also made life easier when it came to maintaining and debugging.

I also attempted to emulate the workflow of an actual software team, based on sayings found in professional projects. I outlined things like; which of theirs to put where, how we should group up components and manage state, and how deep in the stack calls for their services would be. That reduced the chance of future disruptions and made a more reliable development environment.

FILE STRUCTURE

```
/CyberSecurity-Management-System
├── backend/ → .NET Core Web API
│   ├── Controllers/
│   │   ├── AuthController.cs ← Login ve Register endpointleri burada
│   │   ├── DevicesController.cs
│   │   └── RoomsController.cs
│   ├── Models/
│   │   ├── User.cs
│   │   └── Device.cs
```

```
├── Room.cs
├── DTOs/
│   ├── LoginDto.cs
│   └── RegisterDto.cs
├── Data/
│   └── ApplicationDbContext.cs
├── Program.cs
├── DeviceManager.csproj
├── appsettings.json
└── Dockerfile
```

Figure 8: Backend structure

```
frontend/
├── src/
│   ├── app/
│   │   ├── layout.tsx ← Genel layout (Header, Footer, nav vb.)
│   │   ├── page.tsx ← Ana sayfa (oda ve cihaz listeleme)
│   │   ├── login/
│   │   │   ├── page.tsx ← Login sayfası
│   │   │   └── register/
│   │   │       ├── page.tsx ← Register sayfası
│   │   │       └── components/ ← Sayfalar ortak kullanılan bileşenler
│   │   │           ├── Header.tsx
│   │   │           ├── RoomList.tsx
│   │   │           ├── DeviceList.tsx
│   │   │           ├── EditPanel.tsx
│   │   │           ├── LoginForm.tsx
│   │   │           └── RegisterForm.tsx
│   │   ├── services/ ← API çağrıları ve servis fonksiyonları
│   │   │   ├── api.ts
│   │   │   ├── authService.ts
│   │   │   ├── rooms.ts
│   │   │   └── devices.ts
│   │   ├── utils/ ← Yardımcı fonksiyonlar (isAdmin kontrolü)
│   │   │   ├── auth.ts
│   │   │   └── styles/
│   │   │       └── globals.css
│   ├── public/ ← Statik dosyalar (resimler, favicon vs.)
│   ├── Dockerfile ← Frontend için Dockerfile
│   ├── package.json
│   └── next.config.js
```

Figure 9: Frontend structure

Date	Supervisor's Name	Signature
30.07.2025	Betül Nur İyican	

Day 12:

A lot of the technology I'd be working with was new to me. So most of the day I was busy on my computer, installing all the software required. Before I started, I made sure to be set up on my development environment including an IDE package managers such as npm, version controls like git, testing tools like Postman and a database tool DBeaver.

Once installed, I required the packages for the project, set up environment and tested configurations. I was able to more easily move into project development after the installation, due to detailed planning I had done ahead of time. This period allowed me to discover a lot of techs, and paved the way for a much easier time in my next stages.

Date	Supervisor's Name	Signature
31.07.2025	Betül Nur İyican	

Day 13:

I began frontend programming with React. I originally targeted user login screens. I made the authentication and main log in screens. Leveraging React component model, I split each of this fn to separate components. I saved time and kept the code cleaner by creating reusable components, especially with list structures.

I used React's state for this. I made sure that every change in data is instantly displayed to avoid performance degradation. Hot reload also helped me in testing my changes right away. A short investigation I did of React in advance let me be more mindful of how to clearly take advantage its strengths. This made the first few pages super easy to put together.

Date	Supervisor's Name	Signature
1.08.2025	Betül Nur İyican	

Day 14:

Today, I focused heavily on front-end development, particularly building both the login and registration interfaces. The forms themselves had a polished look, and I implemented real-time validation to provide immediate feedback; misspelled emails or missing fields were flagged without delay.

Throughout the process, I maintained constant communication with my mentor to anticipate potential issues and verify the app's stability. While I was still figuring out the intricacies of React's state management and the interaction of props, I observed that the data flow remained consistent and correct. Frequent comparisons between my code and the Figma prototypes helped ensure the UI closely aligned with the intended design specifications.

At the end of the day, I showed my mentor what I had done and called it a day.

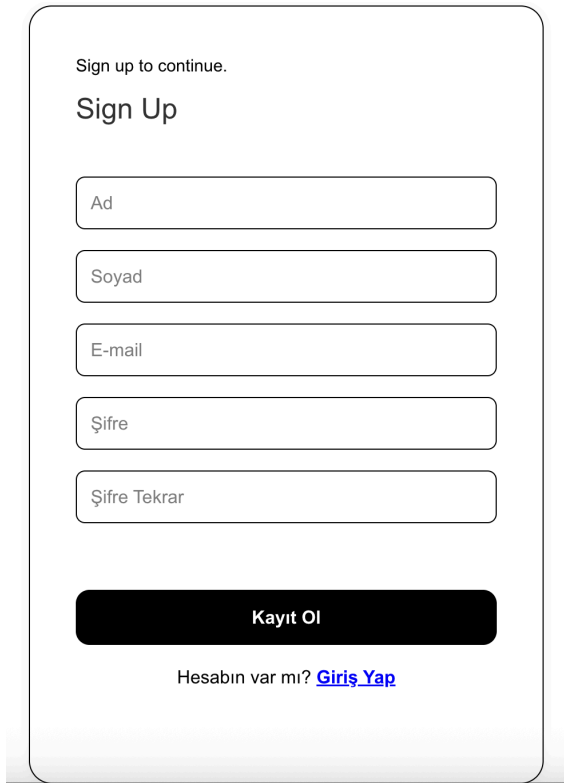
A UI mockup of a registration page. It features a white background with a rounded rectangle containing the text "Sign up to continue." followed by "Sign Up" in bold. Below this are five input fields: "Ad", "Soyad", "E-mail", "Şifre", and "Şifre Tekrar". At the bottom is a black button with "Kayıt Ol" in white text, and a link "Hesabın var mı? [Giriş Yap](#)" below it.

Figure 10: Register page

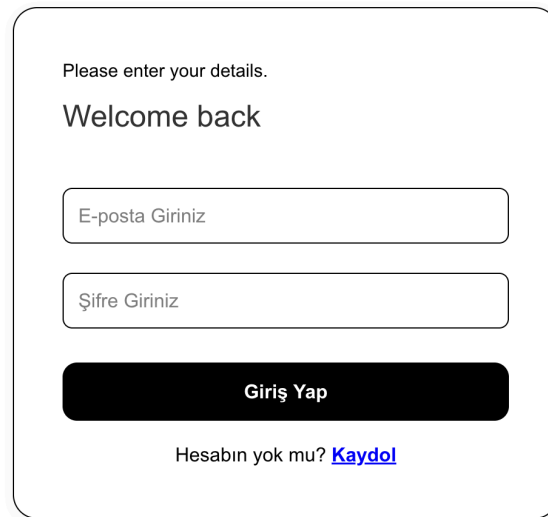
A UI mockup of a login page. It features a white background with a rounded rectangle containing the text "Please enter your details." followed by "Welcome back" in bold. Below this are two input fields: "E-posta Giriniz" and "Şifre Giriniz". At the bottom is a black button with "Giriş Yap" in white text, and a link "Hesabın yok mu? [Kaydol](#)" below it.

Figure 11: Login page

Date	Supervisor's Name	Signature
04.08.2025	Betül Nur İyican	

Day 15:

I went further with other screens of the user panel. I particularly made room and device management pages. In those pages, I added functionality that allows the user to manage devices, obtain a list of rooms, and add new records.

In this stage, I created and reused card parts. Therefore, I used the same part per device or room and avoided code duplication. With this approach, the project became more modular and the interface more uniform. I also ensured responsive design with a layout that works well with screens of different dimensions.

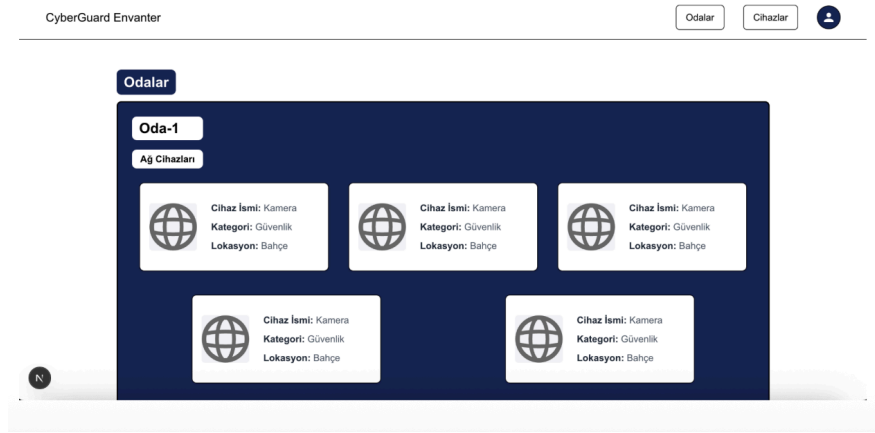


Figure 12: Rooms page

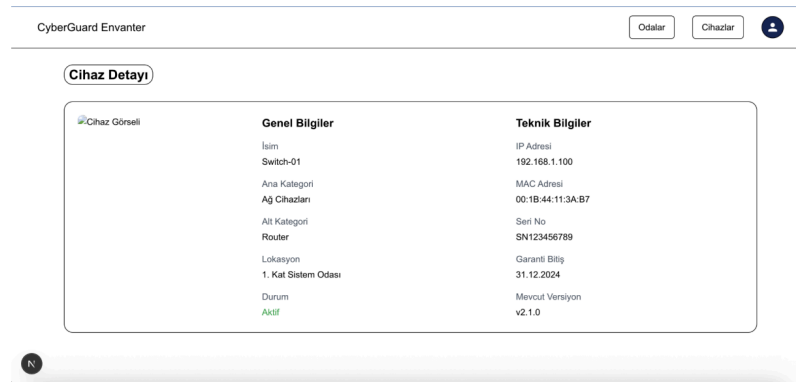


Figure 13: Device detail page

Date	Supervisor's Name	Signature
05.08.2025	Betül Nur İyican	

Day 16:

The day was dedicated to the admin panel. From the pages that I developed at the user end, I developed similar screens at the admin end. But the admin panel required additional buttons and administrative actions. I took special interest in active/deactive buttons and developed mechanisms that allow you to change the status of the device or the end-users.

Component architecture of React simplified the process of transitioning from the UI to the admin interface. With the addition of subtle changes in designs, I integrated privileged admin functionality into the system. Although I needed to implement some minor changes in the plan along the process, the project generally turned out efficiently and promptly.

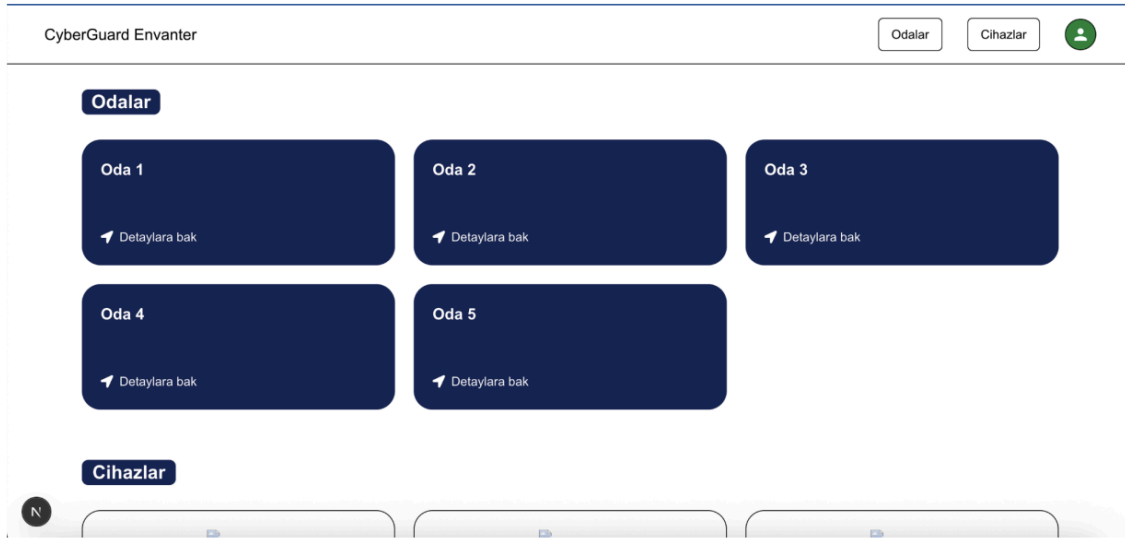


Figure 14: Admin panel rooms

Date	Supervisor's Name	Signature
06.08.2025	Betül Nur İyican	

Day 17:

Having finished the basic development on the front-end aspect, I went on with the back-end. I first established the PostgreSQL database environment with Docker. As the latter happened to be my first encounter with Docker, I first had to get familiar with container logic. I went through and prepared the Dockerfile and docker-compose settings.

Having established the connection with the database, I generated the SQL scripts and implemented them into the project file system. I applied the initial migration and populated the tables into the database. Although this process wasn't real-time, Docker provided a more portable, reliable, and deployable environment. This step became invaluable for me personally, both as a learning process with a new technology and as shaping the professional organization of the project.

Date	Supervisor's Name	Signature
07.08.2025	Betül Nur İyican	

Day 18:

On the server-side, I developed the authentication APIs. I crafted the methods that allow the creation of a registered user and their login and session handling. Here, I utilized the use of standardized data input and output via the DTO (Data Transfer Object) pattern. Using this, both rapid development and uniform data flow were achieved.

First, I tested the APIs with Postman. I learned how to test through a few minute videos as I at first struggled with it. When checking endpoints, I ensured I got correct responses and failed requests were getting correct error responses. I also verified if actually registered users were correct through inspecting the database through DBeaver. It assisted me both in learning API testing and successfully implementing the authentication module.

Date	Supervisor's Name	Signature
08.08.2025	Betül Nur İyican	

Day 19:

Today I used a JWT (JSON Web Token)-based authentication system in my project. Earlier, I only got acquainted with the theory of JWT, but this time I could use it with an actual project. Using it this way helped me realize mechanisms of authentication and authorization used in a project-based setting.

I browsed through the docs and example projects and learned how a JWT is created, at what stage it's sent to the client, and how it's embedded in the header with every request. I needed to research extensively, especially to understand the difference between access tokens and refresh tokens and how to prevent security risks. When I tested the APIs, I utilized Postman and tested the behavior of the token. However, as JWT isn't an exclusive requirement of point-to-point authentication, I could not perform the tests I wished to carry out under certain conditions. My trial and error learning at this point educated me about the importance of secure system architecture.

Date	Supervisor's Name	Signature
11.08.2025	Betül Nur İyican	

Day 20:

I implemented API endpoints of devices and rooms today. I encountered more problems continuing with this phase of work than I first expected, specifically while working with enum types of the PostgreSQL database. Having no prior knowledge of specifying enum types while migrating and managing them at the database level, I had no option other than carrying out extensive research.

To resolve the issues, I conducted a lot of experiments both from the backend code and from the SQL. I also looked at best practices of enum definition from a number of sources and derived a solution that may work with my project. I tested the created APIs using Postman and validated the results from a performance as well as an accuracy standpoint.

I also actively utilized the JWT token during this time. I played around with how the authorization header is sent during requests by sending the token via Postman and how error responses were created with attempts missing the token. It gave a further insight into how JWT actually works in practice.

Date	Supervisor's Name	Signature
12.08.2025	Betül Nur İyican	

Day 21:

I completed the API endpoints that I specially designed for the admin today. I separated the user's and the admin's roles and implemented a role-based access control (RBAC) system. It prevented non-administrative users from carrying out specific tasks.

Apart from authentication of JWT at the backend level, I further implemented role-based control. It means a user could not access a relevant endpoint regardless of how much their token could be valid if they were not possessing the admin role. It created an opportunity of utilizing the most utilized approach of real-world applications of an enterprise type.

During my testing with Postman, I tested each of the user and admin roles separately. I discovered that restrictions were working as expected at the user end and everything was smooth on the admin end. This step enlightened me regarding the importance of not only JWT but also of authorization systems and how they should actually be implemented into the project.

```
// Authorization control
useEffect(() => {
  if (!isAdmin()) {
    router.push('/');
    return;
  }

  loadRooms();

  if (isEditing && deviceId) {
    loadDeviceData(parseInt(deviceId));
  } else {
    setInitialLoading(false);
  }
}, [isEditing, deviceId, router]);
```

Figure 15: Authorization control

```
// Authorization control
useEffect(() => {
  if (!isAdmin()) {
    router.push('/');
    return;
  }

  if (isEditing && roomId) {
    loadRoomData(parseInt(roomId));
  } else {
    setInitialLoading(false);
  }
}, [isEditing, roomId, router]);
```

Figure 16: Authorization control

Date	Supervisor's Name	Signature
13.08.2025	Betül Nur İyican	

Day 22:

Today I began combining the frontend and the backend. I first combined the authentication processes. I checked whether the input data during login and register processes were properly sent into the backend. Then I checked whether the latter were actually processed through the database with the aid of DBeaver.

I also added a manual user I created as the admin into the database. Then I used the user and tried out the process of being directed back into the system's admin interface. When I included the proper scripts into the database, I could spot that the backend operated with the frontend. One of the principal operations that this step carried out involved ensuring that data were both inserted and fetched out of the database correctly.

```
export const authService = {
  // Login function
  login: async (credentials: LoginDto): Promise<AuthResponseDto> => {
    const response = await apiClient.post('/auth/login', credentials);
    return response.data;
  },

  // Register function
  register: async (userData: RegisterDto): Promise<AuthResponseDto> => {
    const response = await apiClient.post('/auth/register', userData);
    return response.data;
  },

  // Token management
  saveToken: (token: string) => {
    localStorage.setItem('auth_token', token);
  },

  getToken: () => {
    return localStorage.getItem('auth_token');
  },
};
```

Figure 17: AuthService page

	123 id	AZ first_name	AZ last_name	AZ email	AZ password	A
1	1	string	string	user@example.com	\$2a\$11\$Z8kPdac19o1F4ln8YoNO u	
2	2	admin	string	user1@example.com	\$2a\$11\$IqYO5Ey5Mr0k6OXeYaQl u	
3	3	adminTest	string	admin@example.com	\$2a\$11\$.H7lmCQDCfTWdQTJucc u	

Figure 18: DBeaver users table

Date	Supervisor's Name	Signature
14.08.2025	Betül Nur İyican	

Day 23:

The devices and rooms related APIs were then integrated in the front-end in a following stage. Especially the data binding was crucial here. I spent some time learning how to show data from the backend on the frontend and update with user interaction.

I also looked at the discrepancies between the admin and user signs on what they saw in the UI. As an example, the admin's login presented a lot more buttons and options than did the standard user login, and provided not much possibility for actions. I tested this and the system gets it right!

This phase taught me the value of communication between frontend and backend, and showed me what role-based UI design looked like in practice.

Date	Supervisor's Name	Signature
15.08.2025	Betül Nur İyican	

Day 24:

Today, after integrating the admin API endpoints beyond my personal alignment into the frontend, I realized that user and admin operations were complete. All the core functions needed by both users and admins were operational.

This process not only helped me integrate but also better understand the inherent applications of authentication and authorization mechanisms. Seeing that users have limited permissions and admins have more rights, I took a step toward understanding the unique logic of social engineering and large-scale, professional projects. Furthermore, the testing I conducted allowed me to see that the system was working stably, giving me greater confidence. In this respect, my project is becoming more mature not only from a technical perspective but also from a project management perspective.

Date	Supervisor's Name	Signature
18.08.2025	Betül Nur İyican	

Day 25:

Today I spent time testing everything on the project. I have individually verified all of the modules that I developed in front-end and back end so that I was assured about no Error at system level. Also I made everything – up to even updating features from the admin panel, device and room managings, system of permissions based on roles etc. - work as it should.

I had been thinking about error situations especially during the tests. For instance, I verified that the system supported correct error messages when a user attempts to login with incorrect password, invalid token usage and when any user who does not have an administrative right is attempting access to admin endpoint. I also did hole workflows in Postman with the API and confirmed if everything is running fine.

I also checked how responsive frontend was on various devices. I checked the performance of pages on mobile, tablet and desktop screen sizes.

Date	Supervisor's Name	Signature
19.08.2025	Betül Nur İyican	

Day 26:

Today, I reviewed the project with my mentor in the office. We discussed the achievements I made during the project, the challenges I encountered, and the solutions I proposed. I also reviewed the project with a few other developers and received feedback.

Date	Supervisor's Name	Signature
20.08.2025	Betül Nur İyican	