



SENG_492/CMPE_492

Graduation Project

Final Report

For

KariyerLAB

TEDU Software/Computer Engineering Department

03.05.2025

Prepared by

Mustafa Karakuş Student ID : 48034466522

Ahmet Hakan Aksoy Student ID : 30649805858

Anıl Çelik Student ID : 29407012522

Advisor: Elif Kurtaran Özbudak

Jury: Tansel Dökeroğlu

Özlem Albayrak

Venera Adanova

1. Introduction.....	4
2. Requirements Details.....	5
2.1 Functional Requirements.....	5
2.1.1 Create CV.....	5
2.1.2 Create Profile.....	5
2.1.3 Apply for the Internship.....	5
2.1.4 Apply for the Job.....	5
2.1.5 Apply for the Scholarship.....	5
2.1.6 Write Comment.....	5
2.1.7 Do Evaluation.....	5
2.1.8 Publish the Posts.....	5
2.1.9 Delete the Posts.....	6
2.1.10 Feedback to User.....	6
2.1.11 Write a Comment.....	6
2.1.12 Login.....	6
2.1.13 Log Out.....	6
2.1.14 Display the Data.....	6
2.2 Non-Functional Requirements.....	6
2.2.1 Performance.....	6
2.2.2 Safety.....	6
2.2.3 Security.....	7
2.2.4 Testing.....	7
2.2.5 Accessibility.....	7
2.2.6 Exception Management.....	7
2.2.7 Availability.....	7
2.2.8 Maintainability.....	8
2.2.9 Usability.....	8
2.2.10 Legal and Regulatory Requirements.....	8
3. Final Architecture and Design Details.....	9
4.1 Class Interfaces.....	12
4.1.1 UI.....	12
4.1.2 Back-End.....	15
4. Development/Implementation Details.....	18
4.1 Client Side.....	18
4.2 Server Side.....	18
4.2 Application Tracking and Feedback System Implementation.....	18
5. Testing Details.....	19
Tested Functionalities.....	19
5.1 Company Register and Login.....	19
5.2 Student Register and Login.....	21
5.3 Scholarship and Internship Announcement.....	24
6. Maintenance Plan and Details.....	28
6.1 Angular and Node.js Package Updates.....	28

6.2	Dependency Management and Security.....	28
6.3	Backup and Recovery.....	28
6.4	Bug Fixes.....	28
7.	Other Project Elements.....	29
7.1	Consideration of Various Factors in Engineering Design.....	29
7.2	Ethical and Professional Responsibilities.....	29
7.3	Judgment and Impacts to Various Contexts.....	30
7.4	Teamwork Details.....	31
7.4.1	Contributing and Functioning Effectively on the Team.....	31
7.4.2	Helping Create a Collaborative and Inclusive Environment.....	31
7.4.3	Taking Lead Role and Sharing Leadership on the Team.....	31
7.4.4	Meeting Objectives.....	31
7.5	New Knowledge Acquired and Applied.....	32
7.5.1	Angular Framework.....	32
7.5.2	Node.js.....	32
7.5.3	Ngrok.....	32
7.5.4	Postman.....	32
7.5.5	JSON Web Token.....	32
8.	Conclusion and Future Work.....	33
8.1	Conclusion.....	33
8.2	Future Work.....	33
8.2.1	Mobile Application.....	33
8.2.2	Global Usage.....	33
8.2.2	Chatbot.....	34
9.	Glossary.....	35
10.	References.....	36

1. Introduction

Today, university students face many difficulties while guiding their careers, and these difficulties often lead them to lose motivation and miss opportunities. Lack of information, especially in accessing opportunities such as internships, jobs, and scholarships, is one of the most fundamental problems of the process. The abundance of advertisements containing fake or incomplete information causes students to waste their time and make wrong choices. Inadequate information about the working environment, culture, or experiences of previous employees in job applications forces students to make critical decisions in the unknown.

In addition, the complexity of the application processes and the fact that they are often suspended without being finalized cause students to become disenchanted with the system. The lack of feedback from companies prevents students from seeing their shortcomings and prevents their development. This process damages not only the motivation of young individuals but also their belief in building a career.

In this period when digitalization is accelerating, job and internship applications are mostly made through online platforms. However, this situation increases competition and makes it difficult for students to access opportunities. In this context, platforms that offer students a reliable, transparent, and functional career management tool are of vital importance. For example, VizyonerGENÇ has filled an important gap by offering students many opportunities in this field. However, KariyerLAB, which was developed by taking inspiration from these initiatives, aims to provide a much more comprehensive response to this need.

KariyerLAB is not only a job or internship application platform; it also offers an integrated system where students can evaluate companies, upload documents, filter advertisements, track their applications, and receive feedback. Thanks to this system, students will be able to see which companies are reliable, be protected from exploitative practices, and plan their careers more consciously.

The biggest goal of this platform is to establish an ecosystem based on mutual trust between students, companies, and scholarship-granting institutions and to provide solutions to the problems encountered in the career journey. KariyerLAB will enable young professionals to start their careers with solid steps, support them, and contribute to their becoming more conscious individuals in the future.

2. Requirements Details

2.1 Functional Requirements

2.1.1 Create CV

Users can create their CVs using the template CV theme. In this way, they can easily deliver their summary information to the employer. Our template CV theme contains all the information that a CV should have. It includes all the necessary information about the user, such as experience, education, age, and hobbies.

2.1.2 Create Profile

Users must have a profile for each user who is an employer or a job seeker. This profile includes summary information such as contact information and personal information about the job seeker. A profile design that summarizes information about the institution is required for the employer. Images can be included in the profile.

2.1.3 Apply for the Internship

The user looking for a job/internship can send information about himself/herself to the employer through the application function. In this way, the employer becomes aware of the user and includes the person in the evaluation process.

2.1.4 Apply for the Job

The user looking for a job/internship can send information about himself/herself to the employer through the application function. In this way, the employer becomes aware of the user and includes the person in the evaluation process.

2.1.5 Apply for the Scholarship

Institutions can also publish scholarship advertisements on their profiles, and these advertisements can be applied for by individual users. In this way, employers are informed about the candidates and carry out the evaluation.

2.1.6 Write Comment

Employers or job seekers can state their satisfaction or dissatisfaction from the moment they are involved in any process until the end of the process. In this way, people share with other users whether the institution has managed this process correctly.

2.1.7 Do Evaluation

Students evaluate the internship, job, or scholarship advertisements they apply for. This evaluation includes the processes, communication, or overall application experience of the company they apply to. Companies will display these evaluations graphically and evaluate them on their own behalf.

2.1.8 Publish the Posts

Companies should be able to publish a new job posting after creating it. The published job posting should be viewable by users according to the relevant category (internship, job, scholarship). Before the job posting is published, the system verifies the job posting by warning about missing or incorrect information.

2.1.9 Delete the Posts

Companies should be able to delete internship, job, and scholarship advertisements they publish on the platform when necessary. This process can be used if the advertisement is no longer valid, contains incorrect information, or if there is a need to update the advertisement.

2.1.10 Feedback to User

Companies that post ads should be able to provide feedback to protect users from fake ads and inform users of suspicious situations. This feature will be implemented to increase the security of applications. Today, there are quite a few fake ads in some applications. We expect companies to respond sensibly to applications made to ads.

2.1.11 Write a Comment

Users can comment on specific posts, such as internships, jobs, or scholarship listings, and view comments from other users. Comments must be in text and have a character limit (e.g., 500 characters). Comments are listed chronologically.

2.1.12 Login

Our login form has two main fields: username and password. If you successfully log in, you will be directed to the home page. During login, the password field is securely masked; that is, it appears as "****" while entering the password.

2.1.13 Log Out

A log-out feature is provided so users can safely end their active sessions and exit the system.

2.1.14 Display the Data

Users who post ads can view their internship, job, and scholarship applications graphically. With metrics such as application numbers, accepted/rejected applications, and application status, this data must be up-to-date.

2.2.15 Get Help From Chatbot

Chatbot is an add-on that aims to speed up the job search process of individual users. By using Chat, users can filter the advertisements for themselves more quickly and thus avoid the hassle of scanning all the advertisements. Chatbot aims to save time.

2.2 Non-Functional Requirements

2.2.1 Performance

The list of applications should be displayed to users in 2 seconds.
Query responses will take under 1 second.

2.2.2 Safety

Data Backup and Recovery: Implementing regular data backups and establishing procedures for data recovery in case of system failures or data loss incidents.

Disaster Recovery Plan: Develop a comprehensive plan to restore system functionality at the end of a natural disaster, cyberattack, or other unforeseen incidents.

Secure Transmission of Sensitive Information: Ensuring that sensitive information such as payment details or personal data is transmitted securely over the network using encryption and secure protocols (e.g., HTTPS).

Access Control and User Permissions: Implementing access control mechanisms to restrict unauthorized access to sensitive data and functionalities within the system.

Error Handling and Logging: Implementing robust error handling mechanisms to gracefully handle unexpected errors and log relevant information for troubleshooting purposes without exposing sensitive data.

2.2.3 Security

Internal Security Requirements: For internal security, the system uses TLS encryption for every data exchange. It performs routine security checks to prevent any risks and limits who can see what based on their job.

JWT Security Requirement: JWT is necessary to authenticate user sessions and securely transport identity information. Measures such as properly signing and encrypting JWTs, ensuring reliable key management, and setting appropriate token expiration times should be taken.

CORS Security Requirement: CORS must be configured correctly in the web application. This involves configuring CORS to allow the browser to load resources from other domains in a manner that does not pose security risks. Incorrect CORS settings may expose the application to risks where attackers could exploit security vulnerabilities.

2.2.4 Testing

The system should be tested regularly under different scenarios. Performance, security, and functionality tests should be performed to ensure that the system works correctly. In addition, improvement studies should be carried out based on user feedback.

2.2.5 Accessibility

It should be fully compatible with mobile devices and different screen sizes. The page's structure and content should be arranged to improve the user experience on different devices.

The menu and page structure should be simple, understandable, and easy to navigate. Users should be able to easily access the information they are looking for.

Drop-down menus should be accessible with the keyboard or using assistive technology. In addition, menus and other navigation elements should work smoothly on mobile devices.

2.2.6 Exception Management

Users' errors should be handled appropriately by the system. If there is an issue, the user should be given instructions on what to do and be presented with informative error messages.

2.2.7 Availability

Our initial target audience will be in Turkey; therefore, bug fixes and updates will be scheduled for midnight to minimize any potential disruption to user access.

Our system will be designed and published to be accessible 24/7, except for updates and developments.

A local network or cellular network connection is required to access the system.

2.2.8 Maintainability

Our subsystems will be developed using microservice architecture, and the subsystems will be loosely coupled to each other so that maintenance can be done easily. Also, during updates, any new modules will be easier to integrate and will not affect other subsystems.

2.2.9 Usability

Users will not need to spend time learning how to use the application. The interface will be intuitive and user-friendly, ensuring seamless and straightforward interactions.

2.2.10 Legal and Regulatory Requirements

Personal Data Protection Law (KVKK): The provisions of KVKK must be applied to a system operating in Turkey.

Use of encryption to protect sensitive user information.

Preparation and approval of legal texts determining the user rights and limits of our site.

The images, software codes, or content used must comply with the licenses.

Compliance with copyright laws.

3. Final Architecture and Design Details

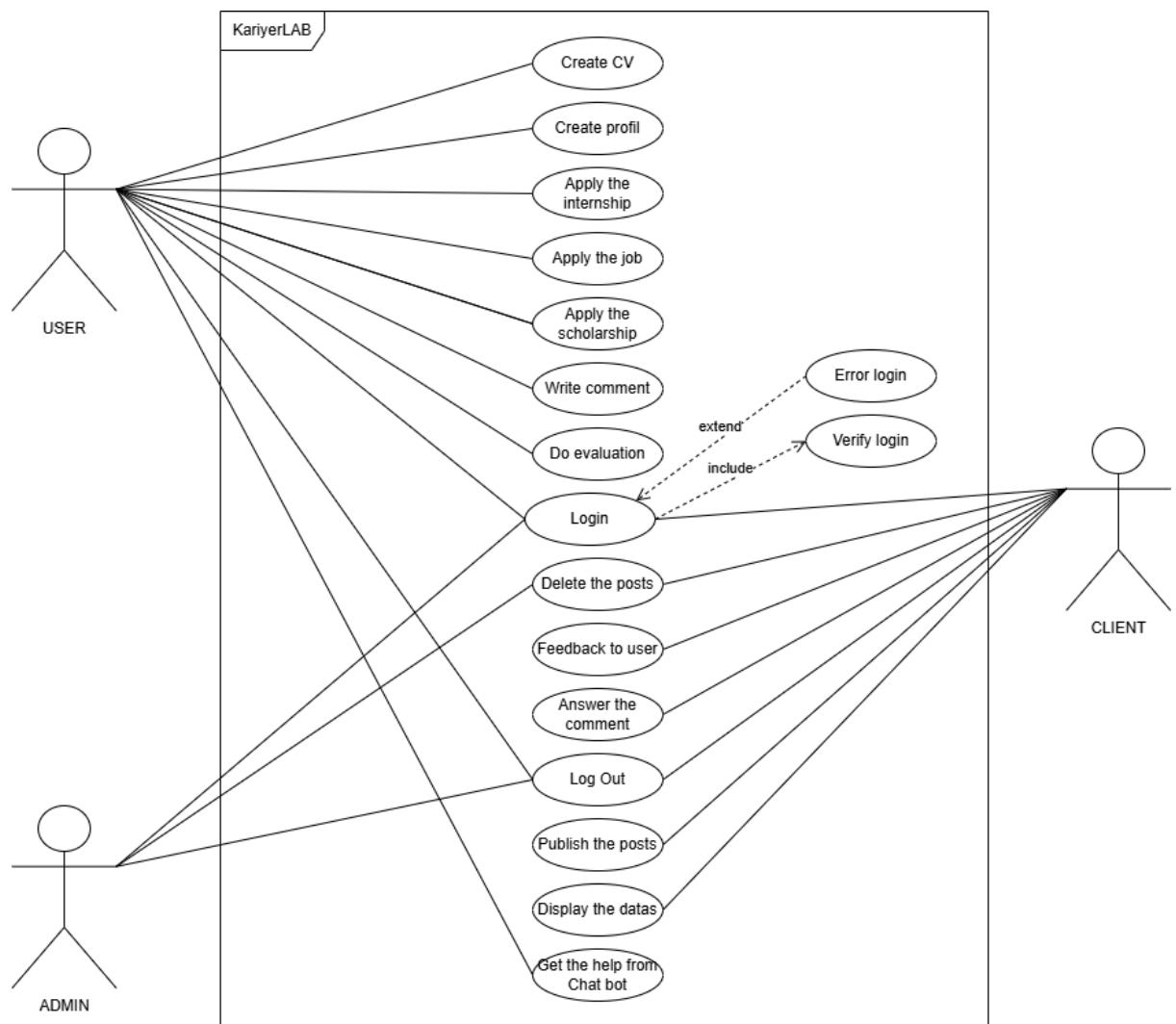


Figure 1: Use Case Diagram

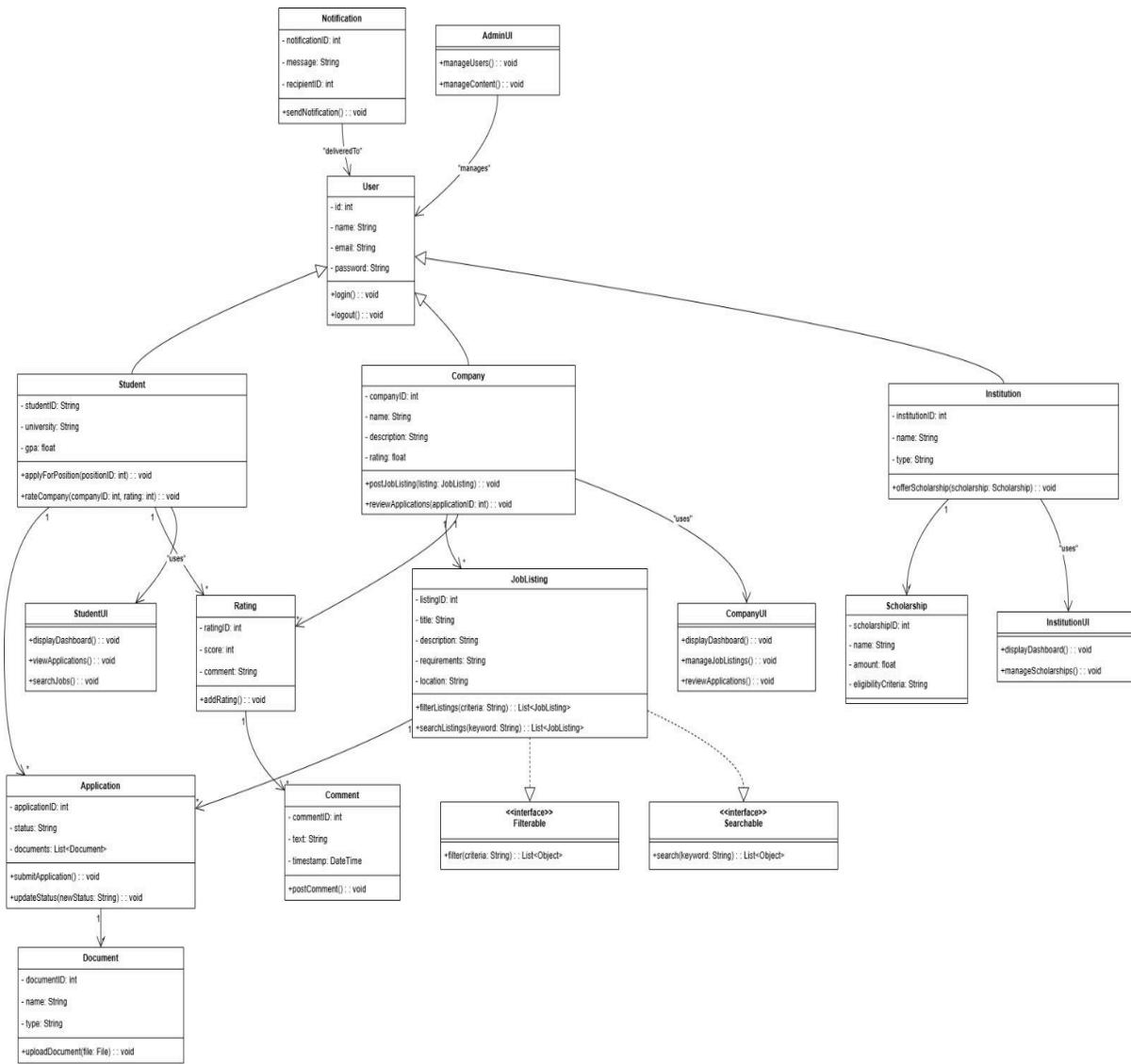


Figure 2: Class Diagram

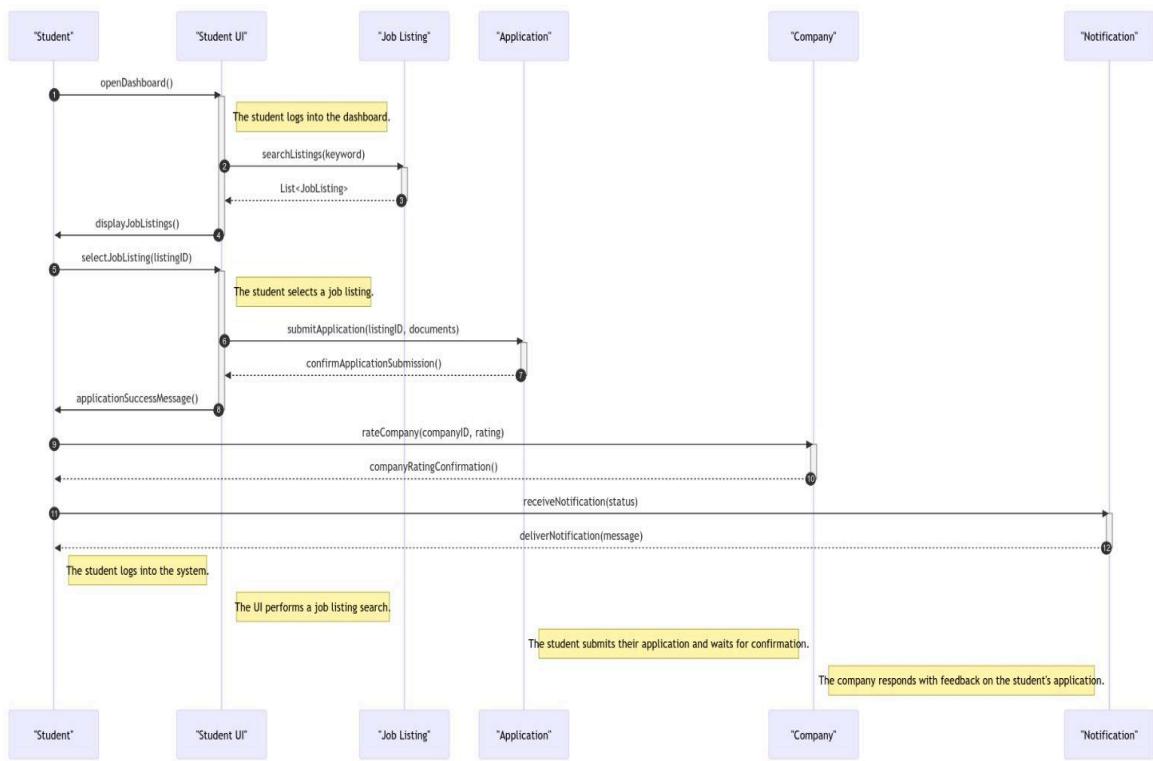


Figure 3: Sequence Diagram

4.1 Class Interfaces

4.1.1 UI

Student
A session of a student is managed by the frontend.
Attributes
<ul style="list-style-type: none">+ email: Student's valid email address+ password: Secure password to be used to log in to the account+ name: Student's name+ surname: Student's surname+ phone: Phone number to be used for communication+ university: University name where the student is registered
Methods
<ul style="list-style-type: none">+ register() registers a student this.apiService.studentRegister(studentData).subscribe({ // Register a new student+ login() { sends login request to api this.apiService.login(this.email, this.password).subscribe() // Student login+ reloadPage(event: MouseEvent): void // reloads the page when the logo is clicked

Company
A session of a company is managed by the frontend.
Attributes
<ul style="list-style-type: none">+ email: The Company's valid email address+ password: Secure password to be used to log in to the account+ companyName: Company's name+ companyAddress: Company's surname+ companyPhone: Phone number to be used for communication+ companySector: Company sector
Methods
<ul style="list-style-type: none">+ reloadPage(event: MouseEvent): void// reloads the page when the logo is clicked+ this.apiService.CompanyRegister(studentData).subscribe({ // Register a new company+ login() { sends login request to api this.apiService.login(this.email, this.password).subscribe()// Company login

Jobs
The session is where companies post job advertisements.
Attributes
<ul style="list-style-type: none"> + companyName: Company name + title: Job title + deadline: Final time to apply job + description: Information about the jobs. + type: Part-time, Full-time
Methods
<ul style="list-style-type: none"> + viewAnnoucementDetail() ----- api requests detailed information about current advertisements by making an http get request + reloadPage(event: MouseEvent): void //Reloads the page when the logo is clicked

Scholarship
The session is where companies post scholar advertisements.
Attributes
<ul style="list-style-type: none"> + scholarName: Name of the scholarship + title: Title or short description + deadline: Deadline + description: Detailed description of the scholarship + type: Type of scholarship (e.g., financial support, abroad, etc.) + organizationName: Name of the institution + organizationEmail: Institution's e-mail address + organizationPhone: Institution's telephone number + organizationAddress: Institution's address + organizationWebsite: Institution's website address
Methods
<ul style="list-style-type: none"> + viewAnnoucementDetail() ----- api requests detailed information about current advertisements by making an http get request + reloadPage(event: MouseEvent): void //Reloads the page when the logo is clicked

Internship
The session is where companies post internship advertisements.
Attributes
<ul style="list-style-type: none"> + internshipName: Name of the internship posting + title: Title or position name + deadline: Deadline for applications + description: Detailed description of the internship + location: Location (hybrid, remote or office) + stipend: Wage or salary information + duration: Internship duration (e.g. 3 months) + eligibility: Application eligibility criteria + skills: Required skills + type: Internship type (mandatory, optional, long-term, etc.) + companyName: Company name + companyEmail: Company email address + companyPhone: Company phone number + companyAddress: Company address + companyWebsite: Company website
Methods
<ul style="list-style-type: none"> + viewAnnoucementDetail() ----- api requests detailed information about current advertisements by making an http get request + reloadPage(event: MouseEvent): void //Reloads the page when the logo is clicked

4.1.2 Back-End

Student
A session for a student is managed by the backend.
Attributes
<ul style="list-style-type: none">+ email: Student's valid email address+ password: Secure password to be used to log in to the account+ name: Student's name+ surname: Student's surname+ phone: Phone number to be used for communication+ university: University name where the student is registered
Methods
<ul style="list-style-type: none">+ exports.register = async (req, res) => { const { email, password, name, surname, phone, university, department } = req.body; } // Register a new student+ exports.login = async (req, res) => { const { email, password } = req.body; } // Student login

Company
A session of a company is managed by the backend.
Attributes
<ul style="list-style-type: none">+ email: The Company's valid email address+ password: Secure password to be used to log in to the account+ companyName: Company's name+ companyAddress: Company's surname+ companyPhone: Phone number to be used for communication+ companySector: Company sector
Methods
<ul style="list-style-type: none">+ exports.register = async (req, res) => const { email, password, companyName, companyAddress, companyPhone, companySector } = req.body; } // Register a new company+ exports.login = async (req, res) => { const { email, password } = req.body; } // Company login

Jobs
The session is where companies post job advertisements.
Attributes
<ul style="list-style-type: none"> + companyName: Company name + title: Job title + deadline: Final time to apply job + description: Information about the jobs. + type: Part-time, Full-time
Methods
<ul style="list-style-type: none"> + exports.getAllJobs = async (req, res) => { // Listed all job opportunity } + exports.getJobById = async (req, res) => { // Listed specific job } + exports.createJob = async (req, res) => { // Create a job }

Scholarship
The session is where companies post scholar advertisements.
Attributes
<ul style="list-style-type: none"> + scholarName: Name of the scholarship + title: Title or short description + deadline: Deadline + description: Detailed description of the scholarship + type: Type of scholarship (e.g., financial support, abroad, etc.) + organizationName: Name of the institution + organizationEmail: Institution's e-mail address + organizationPhone: Institution's telephone number + organizationAddress: Institution's address + organizationWebsite: Institution's website address
Methods
<ul style="list-style-type: none"> + exports.getAllScholar = async (req, res) => { // Listed all scholarship opportunity } + exports.getScholarById = async (req, res) => { // Listed specific scholarship } + exports.createScholar = async (req, res) => { // Create a scholarship }

Internship
The session is where companies post internship advertisements.
Attributes
<ul style="list-style-type: none"> + internshipName: Name of the internship posting + title: Title or position name + deadline: Deadline for applications + description: Detailed description of the internship + location: Location (hybrid, remote, or office) + stipend: Wage or salary information + duration: Internship duration (e.g. 3 months) + eligibility: Application eligibility criteria + skills: Required skills + type: Internship type (mandatory, optional, long-term, etc.) + companyName: Company name + companyEmail: Company email address + companyPhone: Company phone number + companyAddress: Company address + companyWebsite: Company website
Methods
<ul style="list-style-type: none"> + exports.getAllInternship = async (req, res) => { // Listed all internship opportunity } + exports.getInternshipById = async (req, res) => { // Listed specific internship } + exports.createInternship = async (req, res) => { // Create a internship }

4. Development/Implementation Details

4.1 Client Side

KariyerLAB's user interface was developed using the Angular framework. Modern, user-friendly, and accessible design principles were adopted in the interface, allowing students to quickly access the information they needed. Angular components were developed in a modular manner with the TypeScript language, and errors were detected at an early stage by testing the application on the browser. During the UI development process, design prototypes prepared via Canva were referenced, and the appearance and functionality of the screens were discussed in team meetings and continuously improved.

4.2 Server Side

The server side was created using Node.js and Express.js frameworks. MongoDB was preferred as the database. The backend team progressed simultaneously with the UI team and ensured harmony in the integration processes. Postman was used for API tests, while Ngrok was used for better collaboration with the frontend team. A user table was used to hold user information for the authentication process, and the JSON Web Token (JWT) structure was preferred for authorization purposes. The JWT is signed at user login and stored on the client side, and the user's rights in the system are determined with a certain validity period.

4.2 Application Tracking and Feedback System Implementation

One of the most important features offered by KariyerLAB is that users can track their job or internship applications through the system. In this system, the status of the applications (pending, under review, rejected, accepted) is dynamically updated and sent to the user as a notification. At the same time, a comment and scoring infrastructure has been established so that companies can send feedback to applicants. In this way, students can see their shortcomings and contribute to their development. All these processes are managed by Node.js based backend services and presented to the user with an understandable interface on the Angular frontend.

5. Testing Details

During the testing phase, we verified both positive and negative scenarios for several key features of the application using Postman. Each functionality was tested with valid and invalid inputs to ensure both correct behavior and proper error handling.

Tested Functionalities

5.1 Company Register and Login

Company accounts were successfully created using valid data.

The screenshot shows the Postman application interface. The top navigation bar includes 'Home', 'Workspaces', 'Explore', 'Sign In', and 'Create Account'. The main area displays a 'History' section with a list of API requests. A specific POST request to 'https://8139-94-54-22-245.ngrok-free.app/api/company/register' is selected. The 'Body' tab is active, showing a JSON payload:

```
1 {  
2   "email": "company234@example.com",  
3   "password": "strongpassword456",  
4   "CompanyName": "Acme Corp",  
5   "CompanyAddress": "123 Main St",  
6   "CompanyPhone": "9876543210",  
7   "CompanySector": "Technology"  
8 }
```

The response status is '201 Created' with a time of '218 ms' and a size of '301 B'. The response body contains the message: "message": "Company registered successfully". The bottom of the window shows the Windows taskbar with various pinned icons and the date/time '3.05.2025 13:22'.

Login functionality was tested with correct and incorrect credentials.

The screenshot shows the Postman application interface. A POST request is being made to `https://8139-94-54-22-245.ngrok-free.app/api/company/login`. The request body is JSON, containing `{"email": "company234@example.com", "password": "strongpassword456"}`. The response status is 200 OK, with a response body containing a token: `"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MTV1ZTZhMDhZTQ3NDFkNDZkN2Q4YiIsImhdCI6MTc0NjI3MDc0MiwiZXhwIjoxNzQ2MjC0MzQyfQ.JKpNYPana2Vbh01zqjxtQewGJ54pNr4z17w8k0TsxvM"`.

In cases of incorrect passwords or non-existent accounts, appropriate error messages (e.g., "Invalid credentials") were returned as expected.

The screenshot shows the Postman application interface. A POST request is being made to `https://8139-94-54-22-245.ngrok-free.app/api/company/login`. The request body is JSON, containing `{"email": "company234@example.com", "password": "strongpassword45"}`. The response status is 400 Bad Request, with a response body containing the message: `"message": "Invalid credentials"`.

POST https://8139-94-54-22-245.ngrok-free.app/api/company/login

```

1 "email": "company24@example.com",
2 "password": "strongpassword45"

```

Status: 400 Bad Request Time: 139 ms Size: 291 B Save Response

Body

Pretty Raw Preview Visualize JSON

```

1
2 "message": "Company not found"
3

```

Create collections in Postman

Windows'u Etkinleştir Windows'u etkinleştirme için Ayarlar'a gidin.

5.2 Student Register and Login

Student users were able to register and create profiles.

POST https://8139-94-54-22-245.ngrok-free.app/api/student/register

```

1 "email": "mustafa42@example.com",
2 "password": "123456",
3 "name": "Mustafa",
4 "surname": "Karakuyu",
5 "phone": "1234567890",
6 "university": "Example University",
7 "department": "Computer Science"
8
9

```

Status: 201 Created Time: 455 ms Size: 301 B Save Response

Body

Pretty Raw Preview Visualize JSON

```

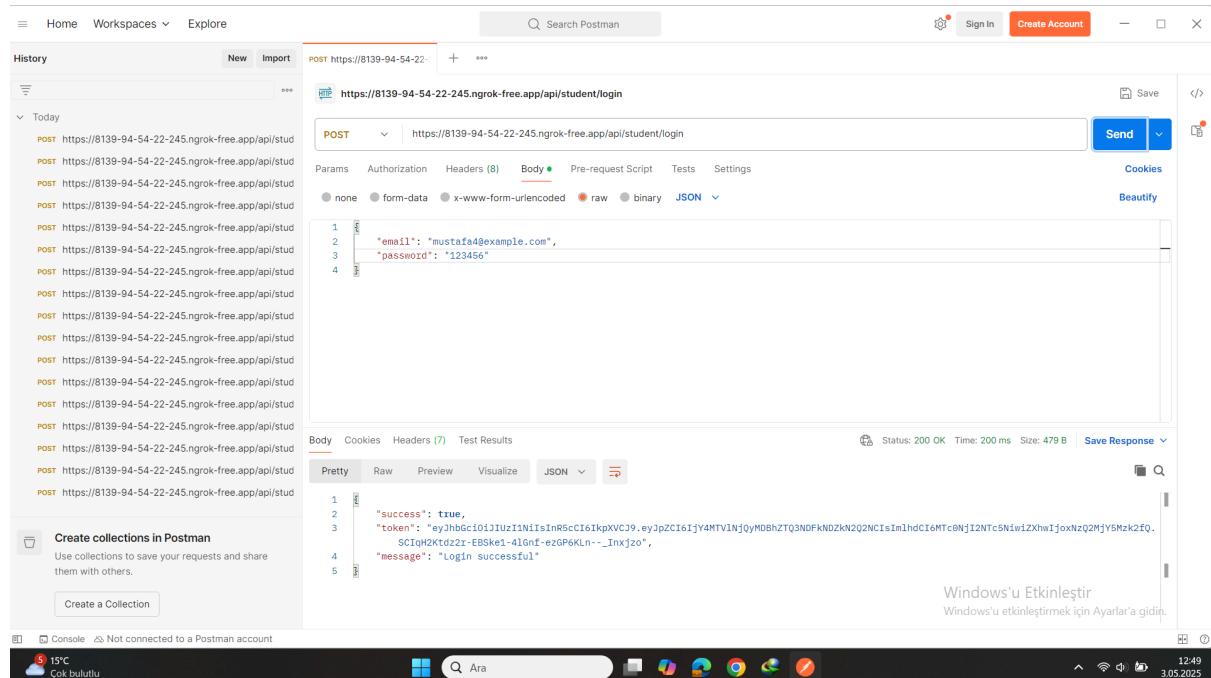
1
2 "message": "Student registered successfully"
3

```

Create collections in Postman

Windows'u Etkinleştir Windows'u etkinleştirme için Ayarlar'a gidin.

Login functionality was confirmed to return tokens and success responses for valid credentials.



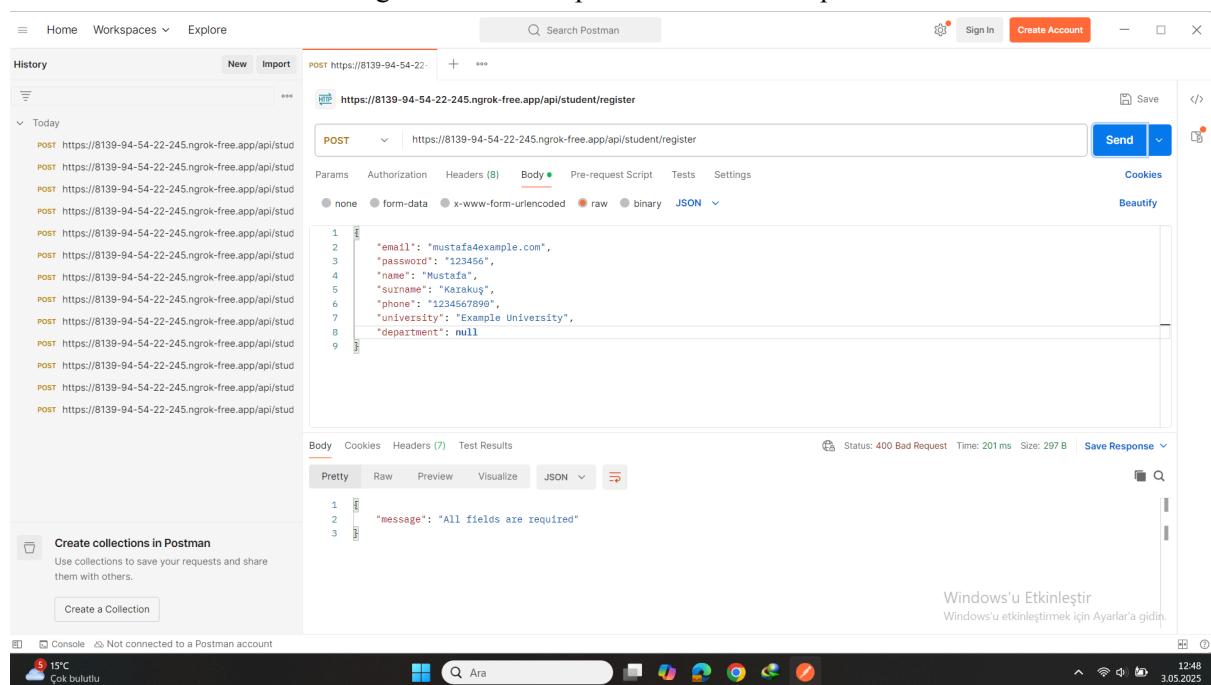
The screenshot shows the Postman application interface. A POST request is being made to <https://8139-94-54-22-245.ngrok-free.app/api/student/login>. The request body is JSON, containing:

```
1 "email": "mustafa4@example.com",
2 "password": "123456"
```

The response status is 200 OK, with a message indicating a successful login:

```
1 "success": true,
2 "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4MTV1NjQyMDlhZTQ3NDdkNDZkN2Q2NCIsImhlhdCI6Mtc0NjI2NTc5NiwiZXhwIjoxNzQ2MjY5Mzk2fQ. SC1Qh2Ktd2r-EBSk1-4lGn-e2GP6KLn--Inxj20",
3 "message": "Login successful!"
```

Student users weren't able to register and create profiles without a requirement.



The screenshot shows the Postman application interface. A POST request is being made to <https://8139-94-54-22-245.ngrok-free.app/api/student/register>. The request body is JSON, containing:

```
1 "email": "mustafa4@example.com",
2 "password": "123456",
3 "name": "Mustafa",
4 "surname": "Karakaya",
5 "phone": "+1234567890",
6 "university": "Example University",
7 "department": null
```

The response status is 400 Bad Request, with a message indicating that all fields are required:

```
1 "message": "All fields are required"
```

POST https://8139-94-54-22-245.ngrok-free.app/api/student/register

```

1 "email": "mustafa4@example.com",
2 "password": "1234",
3 "name": "Mustafa",
4 "surname": "Karakus",
5 "phone": "1234567890",
6 "university": "Example University",
7 "department": "Computer Science"
8
9

```

Status: 400 Bad Request Time: 211 ms Size: 317 B Save Response

Create collections in Postman
Use collections to save your requests and share them with others.

Create a Collection

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Console Not connected to a Postman account

5 HAT - BAS 7 saat içinde

Ara 12:45 3.05.2025

Student users weren't able to log in without a true password.

POST https://8139-94-54-22-245.ngrok-free.app/api/student/login

```

1 "email": "mustafa4@example.com",
2 "password": "12345677"
3
4

```

Status: 200 OK Time: 201 ms Size: 300 B Save Response

Create collections in Postman
Use collections to save your requests and share them with others.

Create a Collection

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Console Not connected to a Postman account

9 15°C Çok bulutlu Ara 12:50 3.05.2025

Tests with incorrect passwords returned meaningful error messages.

The screenshot shows the Postman interface with a failed API registration test. The request URL is `https://8139-94-54-22-245.ngrok-free.app/api/student/register`. The response status is 400 Bad Request, with the message "Password must be at least 6 characters long".

```
1 "email": "mustafa42@example.com",
2 "password": "1234",
3 "name": "Mustafa",
4 "surname": "Karakus",
5 "phone": "1234567890",
6 "university": "Example University",
7 "department": "Computer Science"
8
9
```

Body Cookies Headers (7) Test Results

Status: 400 Bad Request Time: 211 ms Size: 317 B Save Response

Pretty Raw Preview Visualize JSON

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

Windows'u Etkinleştir Windows'u etkinleştirmek için Ayarlar'a gidin.

Console Not connected to a Postman account

HAT - BAS 7 saat içinde

Ara 12:45 3.05.2025

5.3 Scholarship and Internship Announcement

Users were able to post scholarship and internship announcements with proper authorization headers.

The screenshot shows the Postman interface with a successful scholar creation test. The request URL is `https://8139-94-54-22-245.ngrok-free.app/api/scholar/create/scholar`. The response status is 201 Created, with the message "Scholar created successfully".

```
1 "scholarName": "Yüksek Başarı Bursu",
2 "title": "Üniversite Öğrencileri İçin Burs",
3 "deadline": "2025-06-30T23:59:00Z",
4 "description": "Akademik başarı gösteren öğrencilerle yönelik karınlıksız burs programı.",
5 "type": "Karılıksız",
6 "organisationName": "Eğitim Vakfı",
7 "organisationEmail": "info@egitimvakfi.org",
8 "organisationPhone": "02121234567",
9 "organisationAddress": "Atatürk Caddesi No:123, İstanbul",
10 "organisationWebsite": "https://www.egitimvakfi.org"
11
12
```

Body Cookies Headers (7) Test Results

Status: 201 Created Time: 125 ms Size: 826 B Save Response

Pretty Raw Preview Visualize JSON

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

Windows'u Etkinleştir Windows'u etkinleştirmek için Ayarlar'a gidin.

Console Not connected to a Postman account

VIL - OSÅ 5 saat içinde

Ara 12:56 3.05.2025

POST https://8139-94-54-22-245.ngrok-free.app/api/internship/create/internship

```

1 "InternshipName": "Backend Developer Stajı",
2 "title": "React Developer Intern",
3 "deadline": "2025-07-15T23:59:00Z",
4 "description": "Web tabanlı projelerde görev alacak, React bilgisi olan stajyerler arıyoruz.",
5 "location": "remote",
6 "stipend": "5000 TL",
7 "duration": "3 ay",
8 "eligibility": "Üniversite 3. veya 4. sınıf öğrencisi olmak",
9 "skills": "React, JavaScript, HTML, CSS",
10 "type": "mandatory",
11 "companyName": "Tech Solutions",
12 "companyEmail": "h@techsolutions.com",
13 "companyPhone": "02123456789"
14

```

Status: 201 Created Time: 142 ms Size: 963 B Save Response

Situations where error messages are received in some scholarship and internship advertisements.

POST https://8139-94-54-22-245.ngrok-free.app/api/internship/create/internship

```

1 "title": "React Developer Intern",
2 "deadline": "2025-07-15T23:59:00Z",
3 "description": "Web tabanlı projelerde görev alacak, React bilgisi olan stajyerler arıyoruz.",
4 "location": "remote",
5 "stipend": "5000 TL",
6 "duration": "3 ay",
7 "eligibility": "Üniversite 3. veya 4. sınıf öğrencisi olmak",
8 "skills": "React, JavaScript, HTML, CSS",
9 "type": "mandatory",
10 "companyName": "Tech Solutions",
11 "companyEmail": "h@techsolutions.com",
12 "companyAddress": "Levent Mah. İnnovasyon Cd. No:45, İstanbul",
13 "companyWebsite": "https://www.techsolutions.com"
14

```

Status: 500 Internal Server Error Time: 135 ms Size: 360 B Save Response

1 "message": "Internship validation failed: companyPhone: Path 'companyPhone' is required."

(CompanyPhone error)

POST https://8139-94-54-22-245.ngrok-free.app/api/internship/create/internship

```

1 "InternshipName": "Backend Developer Stajı",
2 "title": "React Developer Intern",
3 "deadline": "2025-07-15T23:59:00Z",
4 "description": "Web tabanlı projelerde görev alacak, React bilgisi olan stajyerler arıyoruz.",
5 "location": "remote",
6 "stipend": "5000 TL",
7 "duration": "3 ay",
8 "eligibility": "Üniversite 3. veya 4. sınıf öğrencisi olmak",
9 "skills": "React, JavaScript, HTML, CSS",
10 "type": "mandatory",
11 "companyName": null,
12 "companyEmail": "h@techsolutions.com",
13 "companyPhone": "02123456789",
14 "companyAddress": "Levent Mah. İnnovasyon Cd. No:45, İstanbul",
15 "companyWebsite": "https://www.techsolutions.com"
    
```

Status: 500 Internal Server Error Time: 313 ms Size: 358 B Save Response

Create collections in Postman

Windows'u Etkinleştir Windows'u etkinleştirmek için Ayarlar'a gidin.

(Internship CompanyName)

POST https://8139-94-54-22-245.ngrok-free.app/api/internship/create/internship

```

1 "title": "React Developer Intern",
2 "deadline": "2025-07-15T23:59:00Z",
3 "description": "Web tabanlı projelerde görev alacak, React bilgisi olan stajyerler arıyoruz.",
4 "location": "remote",
5 "stipend": "5000 TL",
6 "duration": "3 ay",
7 "eligibility": "Üniversite 3. veya 4. sınıf öğrencisi olmak",
8 "skills": "React, JavaScript, HTML, CSS",
9 "type": "mandatory",
10 "companyName": "Tech Solutions",
11 "companyEmail": null,
12 "companyPhone": "02123456789",
13 "companyAddress": "Levent Mah. İnnovasyon Cd. No:45, İstanbul",
14 "companyWebsite": "https://www.techsolutions.com"
    
```

Status: 500 Internal Server Error Time: 137 ms Size: 360 B Save Response

Create collections in Postman

Windows'u Etkinleştir Windows'u etkinleştirmek için Ayarlar'a gidin.

(Internship Company Email)

Submissions with missing required fields or without authentication returned appropriate error responses (e.g., "Unauthorized", "Bad Request").

Retrieving Internship and Scholarship Listings

The GET endpoints for internships and scholarships successfully returned all available listings.

Postman screenshot showing the response for the GET /scholar endpoint. The response body is a JSON array of two objects, each representing a scholarship listing. The first object is for 'Yüksek Başarı Bursu' and the second for 'Akademik Başarı Gösteren Öğrencilere Yönельik Karşılıksız Burs Programı'. Both entries include details like title, deadline, description, organization name, email, phone, address, and website.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

Postman screenshot showing the response for the GET /internship endpoint. The response body is a JSON array of two objects, each representing an internship listing. The first object is for 'Frontend Developer Stajı' and the second for 'React Developer Intern'. Both entries include details like stipend, duration, eligibility, skills, type, company name, email, phone, address, and website.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.

The data format and response structure matched expectations.

6. Maintenance Plan and Details

6.1 Angular and Node.js Package Updates

Angular and Node.js applications are updated with new versions over time. These updates may add new features or make improvements to existing features. Working with older versions may cause the application to behave unexpectedly or cause security vulnerabilities. The development team will regularly check the Angular and Node.js packages to ensure that the platform is updated to the latest stable version.

6.2 Dependency Management and Security

On the Node.js side, npm is used for dependency management. Therefore, all dependencies used in the platform will be regularly updated and checked for security vulnerabilities. If any security vulnerabilities are detected, an update will be made immediately, and the platform will be secured.

6.3 Backup and Recovery

A backup strategy will be implemented for unwanted situations such as data loss. Database backups will be made at regular intervals, and a rapid recovery process can be performed in case of emergency. This will ensure that the platform is constantly accessible.

6.4 Bug Fixes

Once feedback and reports are collected, each bug will be prioritized. Critical bugs (e.g., app crashes or a functional feature not working) will be addressed with high priority. Less urgent bugs will be fixed in later updates

7. Other Project Elements

7.1 Consideration of Various Factors in Engineering Design

The first important factor is the user experience. Having a simple, understandable, and intuitive interface on the platform will allow students to easily adapt to the system. The process of applying for an internship or job should not make the student's daily life difficult; on the contrary, it should be completed quickly and effectively. For this reason, accessibility, mobile compatibility, and reducing process steps were prioritized in the user interface design.

Secondly, data security and reliability were prioritized. The resume, ID, and other documents that users will upload to the system will be stored in encrypted form and shared only with authorized parties. In addition, verification mechanisms and scoring systems will be added to increase the accuracy of the advertisements that companies offer to students. Students will be able to leave evaluations for the institutions they have worked for before; this will ensure the formation of a transparent information ecosystem on the platform.

7.2 Ethical and Professional Responsibilities

One of the main ethical concerns in the development of KariyerLAB is the confidentiality of user data. Since the platform will store sensitive information such as CVs, ID documents, and education records, it is our professional responsibility to ensure that this data is securely encrypted and cannot be accessed by unauthorized parties. Any potential data breach would be a serious breach of user trust and personal information protection regulations (e.g., GDPR).

Another ethical challenge is to ensure fair and accurate company ratings and reviews. Students will be able to leave feedback about their internships or work experiences. However, to prevent abuse or slander, we will implement moderation systems to verify that comments are constructive and do not contain offensive or false statements. Companies will also have the right to respond to comments and will maintain a balanced and fair communication channel.

We also recognize that some companies may post misleading or abusive advertisements. To mitigate this, KariyerLAB will mark vetted companies as verified companies on the platform, and any unethical behavior reported by students will result in an investigation and potential removal from the platform.

To ensure transparency, all users (students, companies, scholarship providers) will agree to the Terms of Use and Privacy Policy before using the app. These agreements will clearly state how user data is processed, what permissions are required, and what type of behavior is expected on the platform. Users will have full control over their data, including the right to delete their account and associated information.

7.3 Judgment and Impacts to Various Contexts

Judgment Description	Impact Level	Impact Description
Impact in Global Context	3	The platform can be localized into multiple languages and adapted for international use, especially in Erasmus-style programs or global internship calls.
Impact in Economic Context	8	It will help students find paid opportunities and companies find talent efficiently, creating mutual economic benefits.
Impact in Social Context	7	The platform increases equal access to job and scholarship opportunities, especially for students from disadvantaged backgrounds.
Impact in Environmental Context	2	Digitalizing application processes reduces paper usage, but the environmental impact remains minimal overall.

7.4 Teamwork Details

7.4.1 Contributing and Functioning Effectively on the Team

Team members developed areas of expertise by being responsible for different components throughout the project. Thanks to the progress reports and code review sessions prepared every week, both information sharing and development processes progressed in a synchronized manner.

7.4.2 Helping Create a Collaborative and Inclusive Environment

In order to increase communication and collaboration within the team, weekly meetings were held via screen sharing via Zoom. Task tracking and progress status were transparently carried out using GitHub Projects. The team adopted an effective branching strategy to maintain personal workspaces while continuing common code integration.

7.4.3 Taking Lead Role and Sharing Leadership on the Team

During the project, leadership tasks were shared cyclically according to each member's contribution to the current process. This system both allowed team members to develop their leadership skills and increased their sense of belonging and responsibility. Especially during integration periods, a certain team member organized the integration of all parts in harmony as the integration leader.

7.4.4 Meeting Objectives

While the project did not fully adhere to traditional Agile methodologies, it adopted a flexible and iterative approach. Instead of fixed-length sprints, the team met weekly to assess progress, review individual contributions, and make joint decisions on new steps. This dynamic planning model, supported by short-term goals and continuous feedback, allowed the team to maintain motivation and effectively overcome obstacles. Regular communication, progressive progress, and a sense of shared responsibility ensured that project goals were achieved on time.

7.5 New Knowledge Acquired and Applied

7.5.1 Angular Framework

We used the Angular Framework for the front-end development of our project. Our team members, who had no prior experience with Angular, were first introduced to Angular's structure, modules, and component-based architecture. The Angular team used the built-in CLI (Command Line Interface) tool to quickly and efficiently create the project structure. In our development process, we focused on creating reusable components, using Angular services to manage business logic, and managing state. The framework's dependency injection and two-way data binding features simplified the development process.

7.5.2 Node.js

For the backend of our application, we chose Node.js, which is suitable for developing scalable and high-performance applications. Initially, we had no experience with Node.js, so we first took the time to understand how Node.js handles a large number of requests simultaneously. We developed our REST API using Express.js as a minimal web framework for Node.js. As we developed, we integrated various packages to manage routing and middleware, which helped us establish an organized and modular structure.

7.5.3 Ngrok

For our local development process, we used Ngrok because we needed to expose our local servers to the internet. This tool allowed us to test our local development environment via a public URL, making it easier to test webhooks, third-party service integrations, and API endpoints. Ngrok made our local server accessible remotely, allowing us to run real-time tests and speed up the development process.

7.5.4 Postman

During the development of our API, we used Postman extensively to test API endpoints and generate documentation. Postman allowed us to send HTTP requests to our server and verify that the responses were consistent with the expected results. We also used it to create automated tests, check for boundary cases, and verify the functionality of our API.

7.5.5 JSON Web Token

We integrated JSON Web Token (JWT) to provide secure user authentication. Initially, we considered traditional session-based authentication methods, but after researching best practices, we chose JWT because it is scalable and stateless. JWT allowed us to securely authenticate users without storing session data on the server. After a successful login, the server generates a token, which is sent to the client and is then used in subsequent requests. This method provided secure communication between the client and the server. We also examined the token expiration and renewal mechanisms to ensure security and a smooth user experience.

8. Conclusion and Future Work

8.1 Conclusion

The KariyerLAB project has been completed as a central platform developed to provide solutions to the basic problems that university students encounter in their career journeys. The application has achieved its goal with its basic functions aimed at providing reliability and transparency in internship, job, and scholarship applications. A system that supports career planning has been created by providing students with the opportunity to make decisions with accurate information, follow application processes, and share experiences.

Several important achievements have been made throughout the development process. First, thanks to the ability of users to rate companies and comment, the platform has gained a structure that deters exploitative practices and encourages corporate accountability. Second, features such as advertisement filtering, application tracking, and document uploading have significantly improved the user experience. Third, the user interface has been designed and tested to be easy to use even for users with limited technical knowledge.

Strong steps have also been taken in terms of privacy and data security; users' data has been processed in local systems and protected in a way that can only be accessed by authorized persons. Feedback from users has shown that the platform is perceived positively, especially in terms of ease of use and reliability of advertisements.

The project development process has been a learning process for team members, where both technical skills and team collaboration and leadership skills have improved.

At the end of this stage, we have defined the current functions of the system as well as the areas that need to be developed in the future. Advanced notification systems, direct integration with institutions, artificial intelligence-based recommendation algorithms, and more comprehensive analysis panels are among the main development topics to be targeted in the later stages of the project.

8.2 Future Work

8.2.1 Mobile Application

The mobile application of the KariyerLAB platform will be developed, allowing users to easily track their applications, notifications and feedback from anywhere they want. In this way, accessibility will be increased and user experience will be improved.

8.2.2 Global Usage

In the future, it is aimed to translate the platform into different languages and provide services to international students and institutions. Thus, KariyerLAB will become a career network not only locally but also on a global scale.

8.2.2 Chatbot

An AI-powered chatbot will be developed that can assist users 24/7 with the application process, document uploading, company reviews, etc. This feature will save time by automating user support services.

9. Glassory

KariyerLAB: Our application name.

REST API: Representational State Transfer (REST) is a stateless architectural style based on a client-server design model.

HTTP Request: HTTP, as an application layer protocol, facilitates the transfer of hypermedia documents.

JSON Web Token: JWT is a compact, self-contained token used for securely transmitting information between parties as a JSON object, often for authentication and authorization.

KVKK: The law that was created to protect users' data. (Kişisel Verilerin Koruma Kanunu)

Admin: The people who are the system owners.

Users (Students): Students who registered in the system

Client (Companies): Companies posting ads on the website.

GDPR: The General Data Protection Regulation (GDPR) is a General data protection regulation enacted by the European Union that regulates the protection of personal data and the privacy rights of individuals.

10. References

- [1] Hosting.com.tr, “What is REST API? REST API Usage and Details,” *Hosting.com.tr Knowledge Base*, [Online]. Available: <https://www.hosting.com.tr/bilgi-bankasi/rest-api/>. [Accessed: 03-May-2025].
- [2] Angular.dev, “Overview - Angular,” *Angular Official Documentation*, [Online]. Available: <https://angular.dev/overview>. [Accessed: 03-May-2025].
- [3] Node.js, “Introduction to Node.js,” *Node.js Documentation*, [Online]. Available: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>. [Accessed: 03-May-2025].
- [4] JWT.io, “Introduction to JSON Web Tokens,” *JWT.io*, [Online]. Available: <https://jwt.io/introduction>. [Accessed: 03-May-2025].
- [5] MongoDB Inc., “Document Data Model,” *MongoDB Manual*, [Online]. Available: <https://www.mongodb.com/docs/manual/core/document/>. [Accessed: 03-May-2025].
- [6] Postman, “Overview,” *Postman Learning Center*, [Online]. Available: <https://learning.postman.com/docs/introduction/overview/>. [Accessed: 03-May-2025].
- [7] TED University Senior Project Reports – High-Level Design, Detailed Design, and Multidisciplinary Delivery Reports