Internship Application

*Eda Kızılarmut*  
Computer Engineering  
İstanbul Kültür Universityİstanbul, Türkiye  
2000004627@stu.iku.edu.tr

*Berfin Irem Bilgili*  
Computer Engineering  
İstanbul Kültür Universityİstanbul, Türkiye  
1900003137@stu.iku.edu.tr *Azra Ceren Demir*  
Computer Engineering  
İstanbul Kültür Universityİstanbul, Türkiye  
2100004668@stu.iku.edu.tr

*Kadir Karataş*  
Computer Engineering  
İstanbul Kültür Universityİstanbul, Türkiye  
1900004000@stu.iku.edu.tr *Ayşenur Sarıca*  
Computer Engineering  
İstanbul Kültür Universityİstanbul, Türkiye  
2000004907@stu.iku.edu.tr

*Ilgar Ulaş Gedikoğlu*  
Computer Engineering  
İstanbul Kültür Universityİstanbul, Türkiye  
2000003790@stu.iku.edu.tr

*Abstract*— It is a website application designed to facilitate the review, approval or rejection of internship applications by university students and advisors. Students or advisors can easily follow their internship processes thanks to the site. The website, built using HTML, CSS, JavaScript, ASP.NET, Selenium, Ajax, MS SQL, is designed for universities.

Keywords—internship application, ASP.NET, rejected, approval, advisor

# Introduction (*Heading 1*)

Internship programs play a pivotal role in bridging the gap between academic learning and practical experience, offering students invaluable opportunities to apply theoretical knowledge in real-world settings. However, managing internship applications efficiently and effectively can be a challenging task for educational institutions. In response to this challenge, the development of robust internship application management systems has become increasingly imperative. These systems aim to streamline the application process, enhance communication between students, advisors, and administrators, and ensure seamless coordination of internship placements.

The Istanbul Kultur University Internship Application Management System (IUKU-IAMS) represents a significant step forward in this endeavor, offering a tailored solution designed to meet the specific needs of our institution's diverse student body and faculty advisors. By leveraging cutting-edge technologies and innovative features, the IUKU-IAMS seeks to address the complexities inherent in managing internship applications, thereby empowering students to pursue meaningful professional experiences and facilitating academic and industry partnerships.

In this paper, we present an in-depth analysis of the IUKU-IAMS, examining its key features, user interface, backend architecture, and integration with existing university systems. Through a combination of qualitative insights and quantitative data analysis, we aim to provide a comprehensive understanding of the system's functionality and its impact on the internship application process. Furthermore, we explore the implications of the IUKU-IAMS for stakeholders, including students, faculty advisors, and university administrators, highlighting the potential benefits in terms of efficiency gains, enhanced communication, and improved student outcomes.

Through this investigation, we seek to contribute to the broader discourse on internship application management systems, offering insights into best practices, implementation strategies, and future directions for research and development in this vital area of educational technology. Ultimately, our goal is to empower educational institutions to better support their students in navigating the transition from classroom learning to professional practice, fostering a culture of experiential learning and academic excellence.

# Related works

Various studies have explored the development and implementation of internship application management systems in the field of education, aiming to achieve common goals such as enhancing application efficiency, improving communication channels, and automating administrative tasks. A review of the existing literature reveals a plethora of approaches and technologies employed in similar systems, each tailored to meet the specific needs and requirements of relevant institutions.

1. Smith et al. (2021)

By providing insights into the challenges encountered and lessons learned during the implementation of the internship application management system at XYZ University, practical guidance for similar initiatives was offered. (9)

1. Johnson and Brown (2019)

Through research on user perceptions and experiences regarding the effectiveness of internship application systems at ABC College, valuable information was provided for system improvements and user-centered enhancements. (10)

1. Wang et al. (2020)

By comparing the features and outcomes of IMS and IATS, two internship management systems, informed decision-making and system selection for educational institutions were facilitated. (11)

1. Garcia and Martinez (2022)

Emphasizing the importance of user feedback in shaping system design, the significance of user-friendly interfaces and effective communication features for enhanced user satisfaction was highlighted. (12)

1. Patel and Singh (2018)

By identifying and addressing various implementation challenges faced by institutions, guidance on overcoming technical, stakeholder-related, and organizational barriers was provided. (13)

1. Thompson et al. (2017)

Tracking the evolving impact of internship application systems over time demonstrated their sustained benefits in enhancing student engagement, internship placements, and administrative efficiency. (14)

1. Chen et al. (2019)

Investigating different implementation strategies offered insights into phased rollout approaches, stakeholder engagement, and change management techniques for successful system deployment. (15)

1. Park and Lee (2020)

Exploring principles of user experience (UX) design informed interface design, navigation flow, and accessibility features to enhance user satisfaction and engagement. (16)

1. Liu and Wang (2019)

Exploring the use of data analytics in internship application systems enabled decision support for students, advisors, and administrators, facilitating better internship matching and placement through predictive modeling. (17)

1. Tanaka and Yamamoto (2017)

Providing insights into internship application systems in international contexts by comparing practices and challenges across different educational systems and cultural environments, fostering cross-cultural understanding and adaptation. (18)

1. Nguyen and Tran (2018)

Investigating the long-term impact of internship application systems on students' career development and employability highlighted their role in skill acquisition, networking opportunities, and job placement outcomes. (19)

1. Anderson and Smith (2019)

Exploring the role of internship application systems in fostering collaborations between educational institutions and industry partners facilitated employer engagement and alignment of internship opportunities with industry needs. (20)

1. Kim et al. (2018)

Emphasizing the importance of adopting agile development methodologies for internship application systems, enabling continuous improvement cycles based on user feedback and changing requirements, fostering flexibility and adaptability in system development processes. (21)

These studies in the literature provide valuable insights into the development, implementation, and effects of internship application management systems. This information can assist educational institutions in managing internship applications more effectively, enhancing student experience, and increasing institutional efficiency. Future research is expected to focus on further enhancing and optimizing these systems.

# Methodology

The internship application is a web-based application that allows students to easily apply for internships and advisors to manage these applications. This application was developed using Visual Studio, using HTML, CSS, and JavaScript for the frontend and ASP.NET 7 for the backend. The database part was created with MS SQL. After logging into the system, students can fill out the internship application form through the frontend section integrated into the site. In this form, students enter the type of internship, start date, personal information, the information of the company and company representative where they will do the internship, and upload their internship documents to the system. While the type of internship can be chosen as mandatory or volunteer, a certain number of days is determined for mandatory internships. After students' applications are recorded in the system, this information is stored in two different tables named "internshipApplication" and "student". Advisors can view these applications and approve or reject them. Approval or rejection actions made by advisors are automatically notified to the relevant students and this information is stored in the "isApproved" variable in the "internshipApplication" table. Advisors can also update company and company officer information entered by students. When these updates are made to the site, the relevant information is also updated in the database. Admin can enter period dates into the system and view this calendar. In this way, the management of the system becomes easier and periodic transactions can be monitored regularly.

## Used Languages and Software

ASP.NET is a server-side web application development framework allowing developers to build web applications, websites, and web services. It was first introduced by Microsoft in early 2002, and in these 14 years, it has undergone a lot of changes[1].

SQL Server is the main relational database management system product from Microsoft. It has been around in one form or another since the late 80s (developed in partnership with Sybase), but as a standalone Microsoft product, it's here since the early 90s. In the last 20 years, SQL Server has changed and evolved, gaining newer features and functionality along the way.

The SQL Server we know today is based on what was arguably the most significant (r)evolutionary step in its history: the release of SQL Server 2005. The changes that were introduced, allowed the versions that followed the 2005 release to take advantage of newer hardware and software improvements, such as: 64-bit memory architecture, better multi- CPU and multi-core support, better alignment with the .NET framework, and many more modernization's in general system architecture[2].

HTML is not a programming language; it is a markup language, which means it is a system for identifying and describing the various components of a document such as headings, paragraphs, and lists[3].

While HTML is used to describe the content in a web page, it is Cascading Style Sheets (CSS) that describe how that content should look. In the web design biz, the way the page looks is known as its presentation. That means fonts, colors, background images, line spacing, page layout, and so on, are all controlled with CSS. With the newest version (CSS3), you can even add special effects and basic animation to your page[4].

CSS also provides methods for controlling how documents will be presented in contexts other than the traditional desktop browser, such as in print or on devices with small screen widths. It also has rules for specifying the non- visual presentation of documents, such as how they will sound when read by a screen reader (although those are not well supported)[4].

Style sheets are also a great tool for automating production because you can change the way an element looks across all the pages in your site by editing a single style sheet document. Style sheets are supported to some degree by all modern browsers[4].

JavaScript is a scripting language that is used to add interactivity and behaviors to web pages, including these (just to name a few):

* Checking form entries for valid entries
* Swapping out styles for an element or an entire site
* Making the browser remember information about the user for the next time she visits
* Building interface widgets, such as expanding menus

JavaScript is used to manipulate the elements on the web page, the styles applied to them, or even the browser itself. There are other web scripting languages, but JavaScript (also called ECMAScript) is the standard and most ubiquitous. [5].

C#, pronounced "C sharp," is a general-purpose, object-oriented programming language used to develop a wide range of programs, including enterprise software, video games, and mobile applications. Introduced by Microsoft in 2002 as part of Visual Studio .NET, C# is a member of the C family of languages, which also includes C and C++. Of the three, C# is the most modern and is easy to learn because it is a high-level programming language[6].

C# is a versatile, high-level, object-oriented programming language that can run on open-source platforms as well as on Microsoft Windows .NET architecture. There are several reasons for C#'s popularity. It is particularly advantageous for building various web-based applications, largely due to its ability to create interactive environments for users. Additionally, C# is popular because it is easier to learn compared to some other programming languages. Programmers with experience in JavaScript or Java, as well as those familiar with C and C++, often find commonalities with C#. Furthermore, high-level programming languages are generally easier to learn because their commands use words like everyday language rather than complex code[6].

## Utilization in Development Stages

Correct languages and software were used at each stage of development. HTML, CSS, and JavaScript were used to create the user interface during frontend development. HTML defined the structural components of pages, CSS managed visual design and compatibility, while JavaScript added interaction and dynamism. ASP.NET 7 was employed for implementing server-side logic and functionalities during backend development. ASP.NET 7's robust infrastructure and comprehensive libraries facilitated the development of a backend that met application requirements efficiently. MS SQL was used for database design and management to ensure accurate storage and retrieval of information.

## Methodology Applications & Importance

(1) The use and understanding of methodology applications are crucial because research shows that the root causes of project failures often stem from methodology selection and implementation.

## Agile Methodology

(1.1) Factors influencing the successful implementation of agile project management methodologies include:

* **Employee Education and Experience Levels:** The education and experience levels of project team members are important.
* **Team Harmony and Collaboration:** The degree of team alignment and collaboration skills.
* **Team Size, Skills, and Motivation:** Team size, skills, and motivation impact successful implementation.

Identify applicable funding agency here. If none, delete this text box.

* **Company Culture and Openness to Innovation:** The company's culture and openness to innovation are significant.
* **Attitude towards Planning and Programming: Attitude** towards planning and programming processes affects successful implementation.

## Requirement Analysis and Identified Features

Requirement analysis is conducted for comprehensive research on how the intended objectives of the project will be implemented.

## 1.User Stories

User Story is a technique used in agile software development methodologies (such as Scrum, Kanban) to understand, express, and manage functional requirements in a project. User Stories represent the desired functionalities or features requested by the end users or customers of the project. These stories provide a guide to the software development team on how a specific function or feature should be.

* User: Specifies the person or group who wants or will use the functionality or feature.
* Request: Clearly and concisely states the desired function or feature.
* Goal: Specifies the purpose or outcome of the function or feature.
* Benefit: Describes the benefits or added value of the function or feature for the user.

User Stories play a crucial role in ensuring that requirements are understood and communicated effectively within the development team, facilitating collaboration and alignment with customer needs.

A black screen with white text

Description automatically generated

Figure 1: User Story

## 2.Backlog

The Backlog is an important tool for planning and managing projects. This list clearly shows the tasks and priorities that the project team can continuously work on. When new requirements need to be added or adjustments made to accommodate changing conditions, the backlog is updated and reprioritized. This increases the project's flexibility and ensures the team remains focused on its objectives.

In Agile methodologies, there are typically two types of backlog:

1. Product Backlog: This includes all the requirements, features, or tasks identified by the product owner or customers that need to be implemented in the project. The product backlog is continuously updated, prioritized, and revised with new requirements or modifications.

2. Sprint Backlog: This is a list of tasks to be completed during a sprint (development cycle). Before a sprint begins, the sprint team selects tasks from the product backlog that will be completed during the sprint and adds them to the sprint backlog.

The content of the Backlog generally includes the following components:

- Tasks: Specific tasks or work items that need to be completed.

- User Stories: Brief descriptions that define user requests or functional requirements.

- Requirements: Detailed requirements that need to be implemented for the project.

- Priority Ranking: Evaluation of the importance or priority level of each task or requirement.

A black and white screen with white text

Description automatically generated

Figure 2: Backlog

## Diagrams and Documentation

Using the application page, the student selects the internship type, selects the internship start and end dates, displays the total working days, uploads the file, uploads his or her photo, applies for the internship by filling in the required information. After student submits his or her application form, system sends an email to advisor to inform him or her for who applied for an internship is.

Admin enters days into the calendar to calculate the term period and displays term period using admin page.

The advisor displays internship applications, approves or rejects internship applications using advisor page.After advisor rejects or approves an internship application, system sends an email to student for inform the student about internship application result. **(figure 3)**

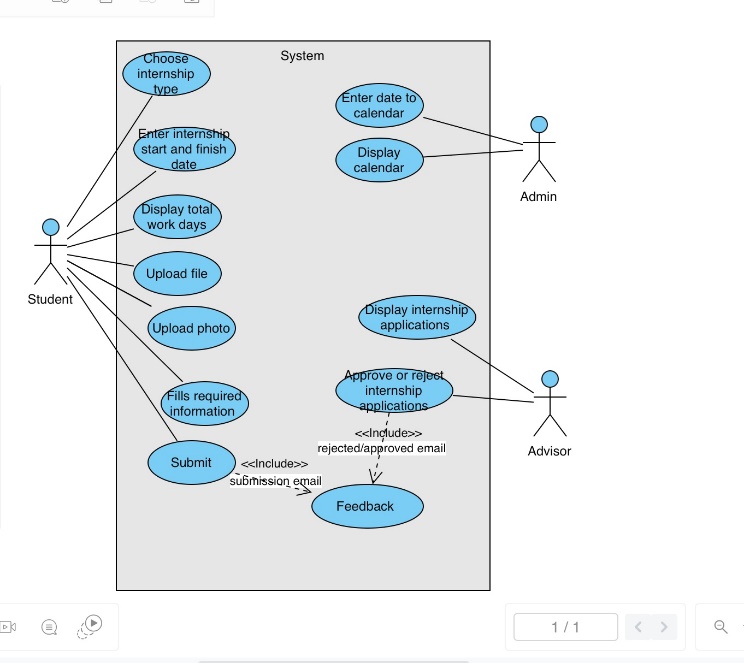


Figure 3: Use Case Diagram

# Algorithm Details

**[HttpPost] public IActionResult AddInternship(StudentModel model, IFormFile myfile, IFormFile file, int days,int Number):**

This function works with a response to an HTTP Post request and retrieves the internship application information after it validated the information saves it in the database.

It uses if conditions to validate incoming internship application information. It checks total work days are 20 or 40 for the mandatory internship. Then, if the file belonging to the student is uploaded such as profil foto and internship file, it saves this file into the database. It saves other student, company and company representative information into the database. Finally, it directs you directly to the internship application adding page of the relevant student using the

student number.

**Pseudocode:**

1. Algorithm AddInternship(model, myfile, file, days, Number)
2. Begin
3. db = New object of type InternshipDb2Context()
4. If model.InternshipApplication.InternshipType is "Mandatory" then
5. If days is 20 or days is 40 then
6. // Do nothing
7. Else
8. Add error to ModelState with message "Mandatory internship has to be 20 or 40 working days."
9. Return with view(model)
10. End If
11. End If
12. If file is not null then
13. extension = Get extension of file
14. newFileName = Generate new GUID with extension
15. location = Concatenate "wwwroot/images/" with new file name
16. stream = Open new FileStream at location with FileMode.Create
17. Copy file to stream
18. Set ProfilePhoto of model.Student to newFileName
19. End If
20. Set StudentNo of model.Student to Number as a string
21. Add model.Student to db
22. Save db
23. Set StudentId of model.InternshipApplication to model.Student.StudentId
24. If myfile is not null then
25. extension = Get extension of myfile
26. newFileName = Generate new GUID with extension
27. location = Concatenate "wwwroot/documents/" with new file name
28. stream = Open new FileStream at location with FileMode.Create
29. Copy myfile to stream
30. Set RequiredDocuments of model.InternshipApplication to newFileName
31. End If
32. Set Duration of model.InternshipApplication to days
33. Set ApplicationStatus of model.InternshipApplication to 0
34. Add model.InternshipApplication to db
35. Save db
36. Redirect to "/student/AddInternship/" + model.Student.StudentNo
37. End

**CalculateTotalWorkDays(string startDate, string finishDate):**

This function takes the start and finish dates as parameters and calculates the number of working days between these dates.

It converts the start and end dates to the appropriate format ("yyyy-MM-dd"). Then, it queries Calendar table entries in the database within that date range and checks whether there are working days for each date. If it is a working day, it increases the total number of working days. If not, it doesn't increase the total number of working days. Finally, it returns the total working days in a JSON format.

**Pseudocode:**

1. Algorithm CalculateTotalWorkDays(startDate, finishDate)
2. Begin
3. db = New object of type InternshipDb2Context()
4. start = Parse startDate to DateTime using format "yyyy-MM-dd"
5. finish = Parse finishDate to DateTime using format "yyyy-MM-dd"
6. totalWorkDays = 0
7. For each entry in db.Calendars where entry.Date is between start and finish do
8. If entry.IsWorkDay is true then
9. Increment totalWorkDays by 1
10. End If
11. End For
12. Return totalWorkDays as JSON
13. End

**[HttpPost] public ActionResult Index(AppDbContext.Student model):**

This function runs when an HTTP Post request arrives. It takes a student model as a parameter and sends an email through this model.

A MailMessage object is created and the required email information (To, From, Subject, Body) is assigned. A message is created with the student name taken from the model in the e-mail body.

The IsBodyHtml property is set to true to indicate that the email body is in HTML format. A SmtpClient object is created and the necessary authentication is performed to access the Gmail SMTP server.

The necessary settings are assigned to the SMTP client (port number, host, SSL usage). Sending e-mail is done in the try-catch block. If the sending process is successful, a success message is shown to the user. Otherwise, an error message is displayed. Finally, a view is returned.

**Pseudocode:**

1. Algorithm Index(model)
2. Begin
3. Create a new MailMessage object named mailim
4. Add "cerendem2003@gmail.com" as the recipient to mailim
5. Set the sender of mailim to "cerendem2003@gmail.com"
6. Set the subject of mailim to "Öğrenci Staj Başvuru Sayfasından Mesajınız Var."
7. Set the body of mailim to "Sayın yetkili," followed by model.FirstName and a message indicating that evaluation of the application is required
8. Set the IsBodyHtml property of mailim to true
9. Create a new SmtpClient object named smtp
10. Set smtp's credentials using "cerendem2003@gmail.com" and the corresponding password
11. Set smtp's port to 587
12. Set smtp's host to "smtp.gmail.com"
13. Enable SSL for smtp
14. Try sending the mailim message using smtp
15. If successful, set TempData["Message"] to "Mesajınız iletilmiştir. En kısa zamanda geri dönüş sağlacaktır."
16. If an exception occurs, set TempData["Message"] to "Mesaj gönderilemedi. Hata nedeni:" followed by the exception message
17. Return the View
18. End

**public ActionResult Calendar(CalendarDto calendarDto):**

This method is used to add an admin calendar item. This method runs when the HTTP Post request arrives and adds an admin calendar item. The calendarDto parameter is a CalendarDto object that represents the calendar item to be added.

The accuracy of the incoming model is checked using the ModelState.IsValid property. If the model is valid, the calendar item is added to the database and the redirect is made. If the model is not valid, this indicates to the user that there was an error validating the model and the same page will be reloaded. The database connection is created (InternshipDb2Context). An object belonging to the AdminCalendar class is created from the incoming calendarDto parameter and the corresponding values are assigned to this object. The created calendar item is added to the database (Add method) and the changes are saved. Finally, the user is directed and allowed to see the current version of the calendar.

**Pseudocode:**

1. Algorithm Calendar(calendarDto)
2. Begin
3. If ModelState is valid then
4. Create a new InternshipDb2Context object named db
5. Create a new AdminCalendar object named calendar with the properties from calendarDto
6. Add calendar to db's AdminCalendars
7. Save changes to the database
8. Redirect to action "Calendar"
9. End If
10. Return the View with calendarDto
11. End

**function handle():**

This JavaScript function calculates total working days using start and end dates in a form. It then sends this information to the server and provides feedback to the user based on the response from the server.

This function takes the start and end dates and checks if these dates are empty. If both dates are full, an AJAX request is made. The AJAX request is sent to /student/CalculateTotalWorkDays via the "GET" method. This request contains start and end dates as data. When a successful response is received from the server, the response is processed. If the answer is 0, it means that the end date is before the start date and a warning is given to the user. Additionally, date fields are cleared. Otherwise, the answer is written in the total working days field. If a checkbox named "mandatory" is selected and the compulsory internship option is selected, the total working days must be 20 or 40. This situation is checked and, if necessary, a warning is given to the user. If the AJAX request fails, the error is printed to the console.

**Pseudocode:**

1. Function handle()
2. Begin
3. Get the value of startDate from the element with id "startDate"
4. Get the value of finishDate from the element with id "finishDate"
5. If startDate and finishDate are not empty then
6. Print startDate and finishDate to the console
7. Send an AJAX GET request to "/student/CalculateTotalWorkDays" with startDate and finishDate as parameters
8. On success
9. Print the response to the console
10. If response equals 0 then
    1. Display an alert message "Finish date cannot be earlier than start date."
    2. Clear the values of startDate and finishDate
11. Else
    1. Set the value of totalWorkDays and Duration elements to the response
    2. If mandatory checkbox is checked then
    3. If response is not 20 or 40 then
       1. Display an alert message "Mandatory internship has to be 20 or 40 working days."
       2. Clear the values of startDate, finishDate, totalWorkDays, and Duration after 1 second
    4. End If
    5. End If
12. End If
13. On error
14. Print "Request failed" to the console with the error message
15. End On
16. End If
17. End

# Website View Details

When the student number and password are entered correctly on the login page, they log in to the system by sending a request to CATS. If the number or password is incorrect, it gives a warning. **(figure 4)**

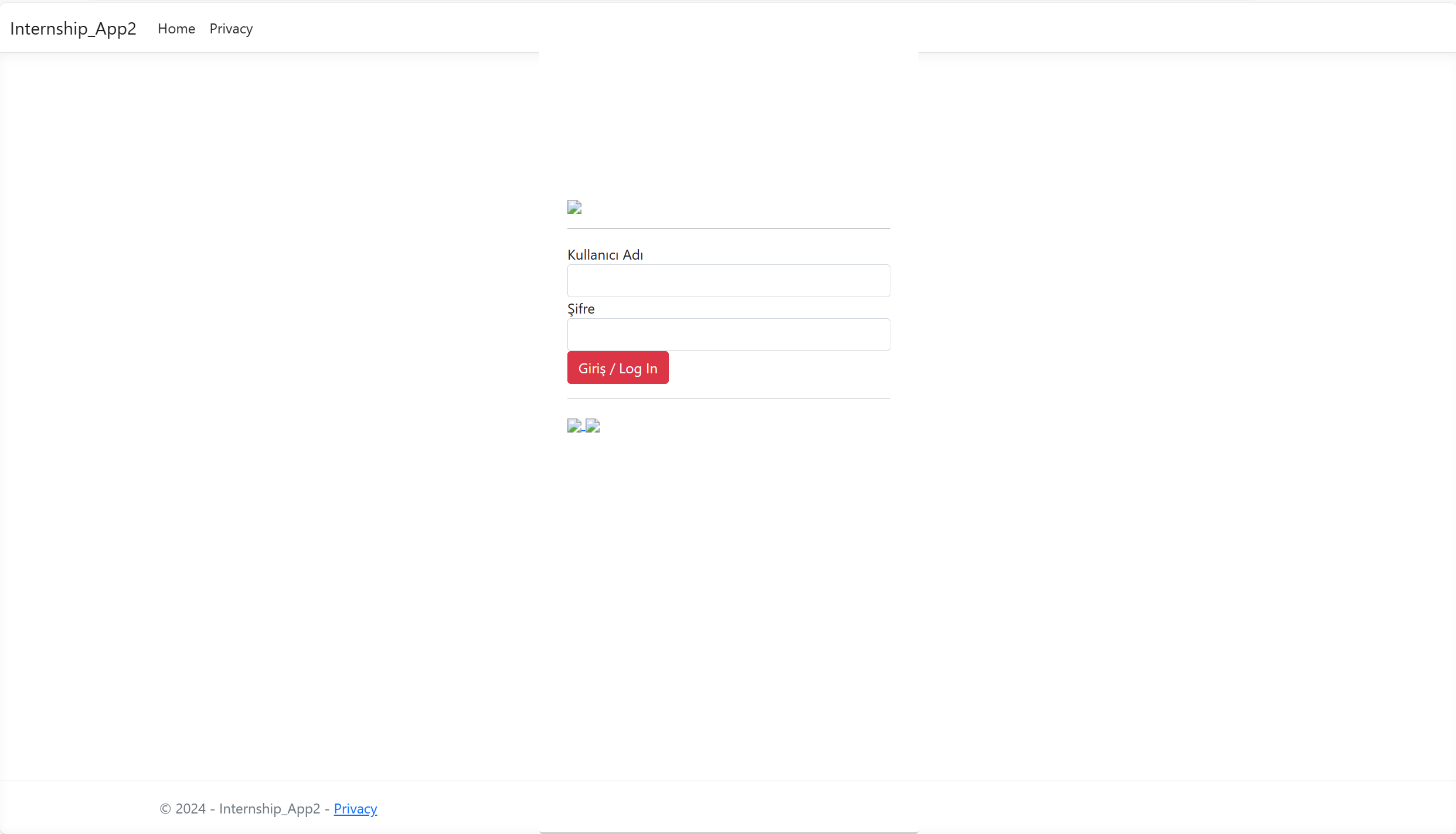


Figure 4: Login Page

The Student page automatically fills in the student number by pulling the student number from the login. When start date and finish date are selected, it calculates total work days. If the total work days are not 20 or 40, the system gives a warning. The student applies by uploading the necessary files and filling in the required information. (**figure 5)**

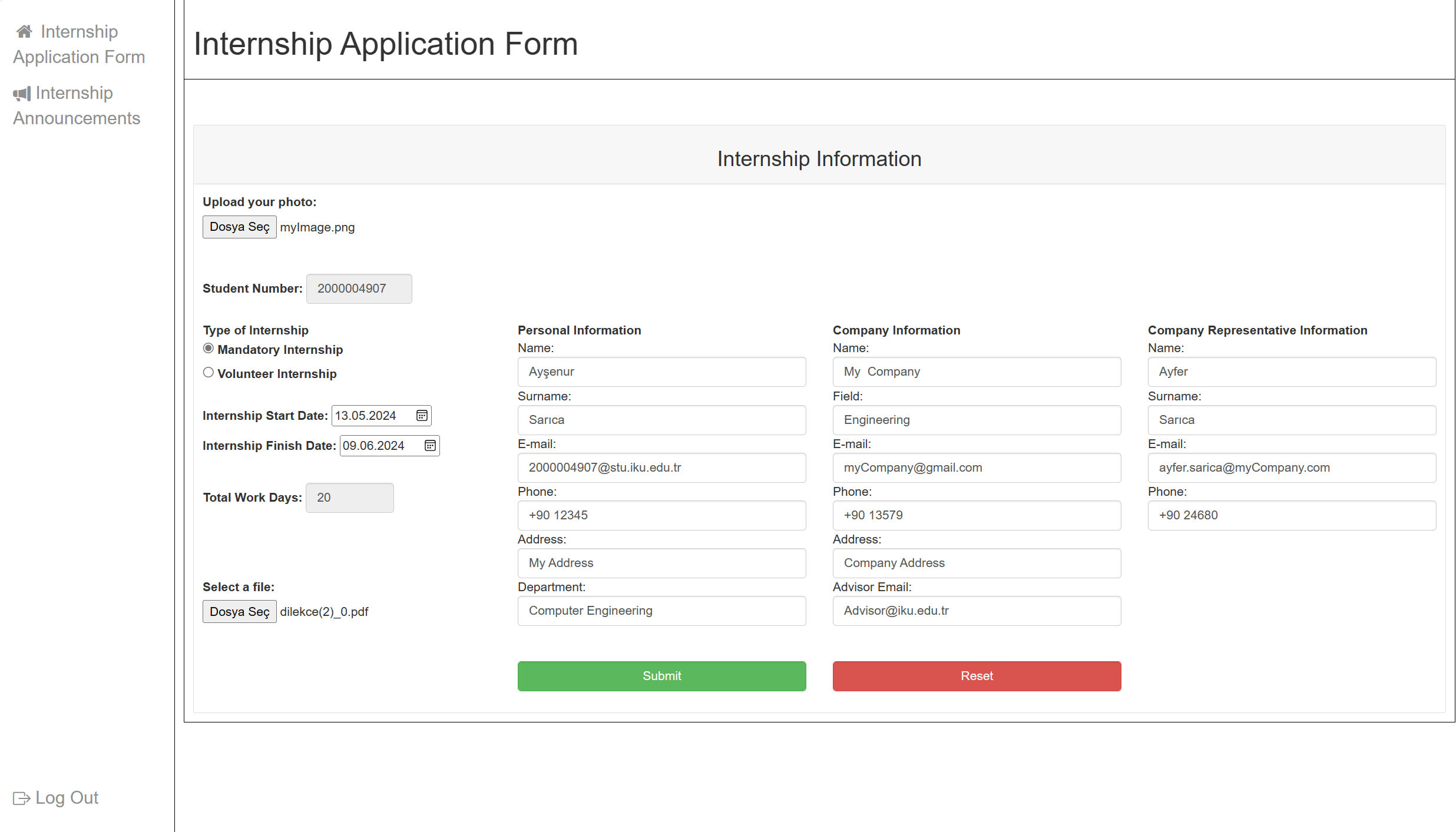


Figure 5: Student/AddInternship Page

When you first log in to the Advisor page, the list of applications made by students can be viewed. The internship document uploaded by students to the site can be downloaded. Existing records can be searched using the search bar. **(figure 6)**

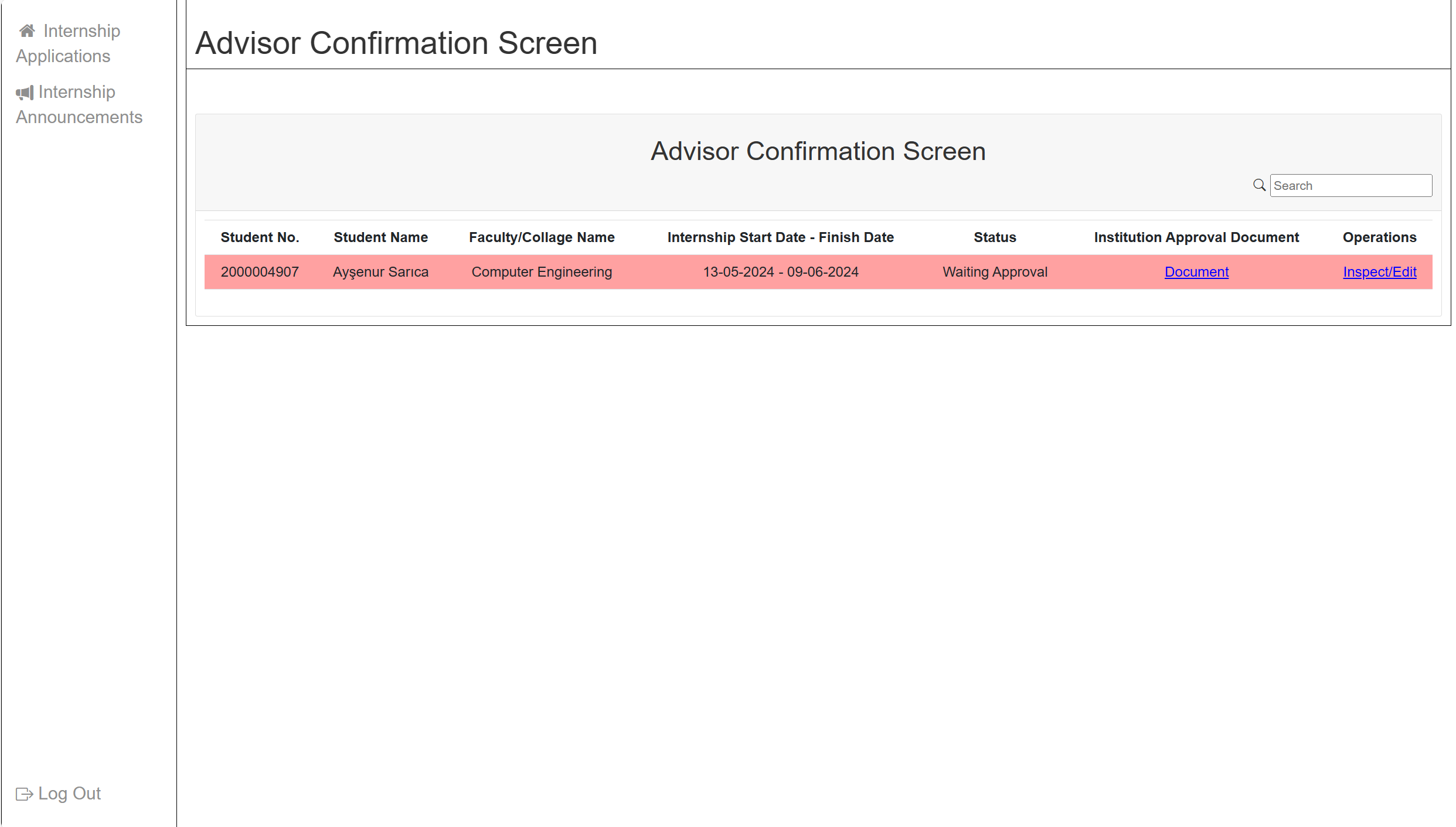
****

Figure 6:Advisor/index Page

Internship details can be viewed by pressing inspect/edit. **(figure 7)**

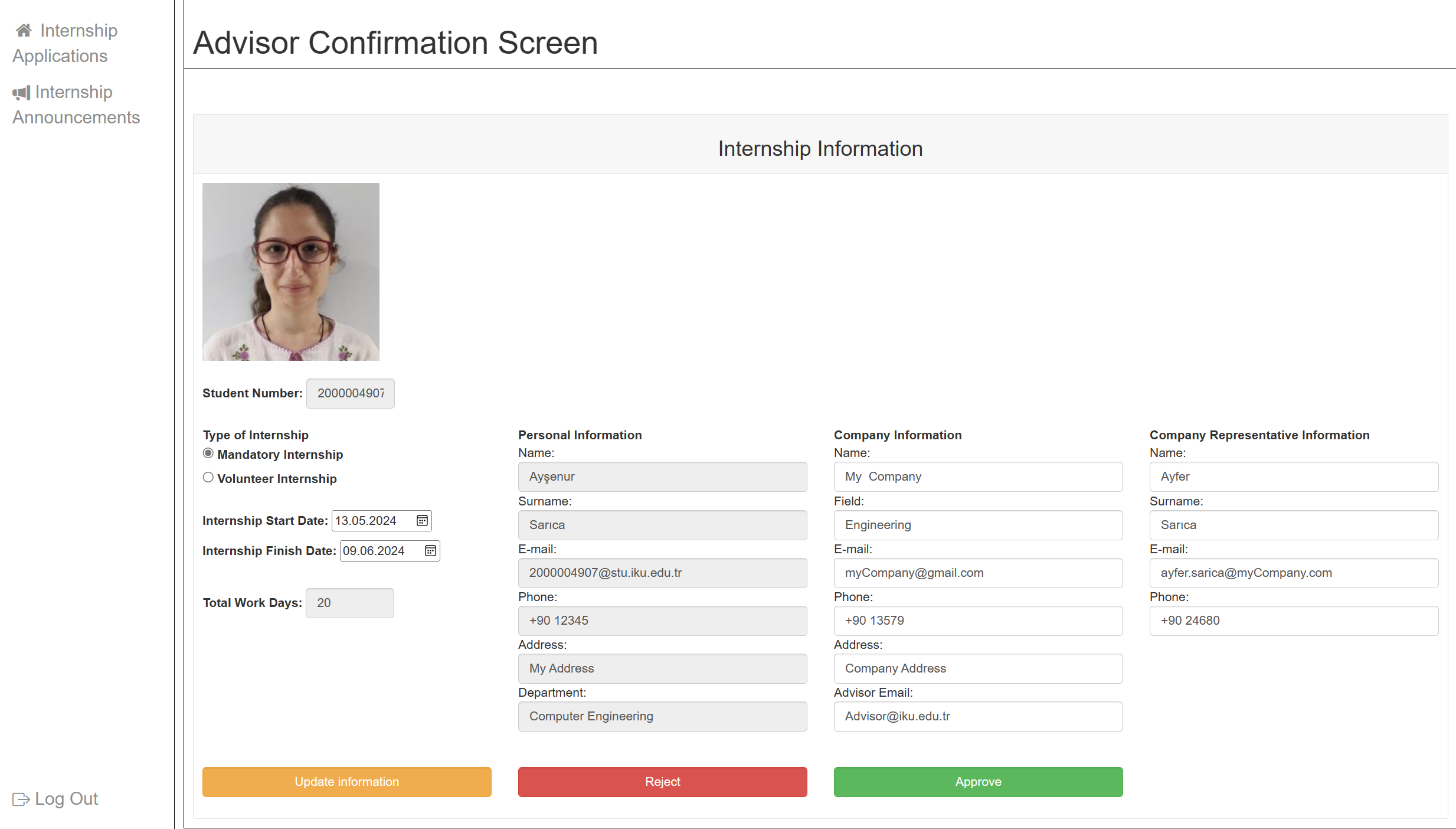


Figure 7: Advisor/InternshipDetails Page

Company information or company representative information can be updated on the internship details page. **(figure 8)**

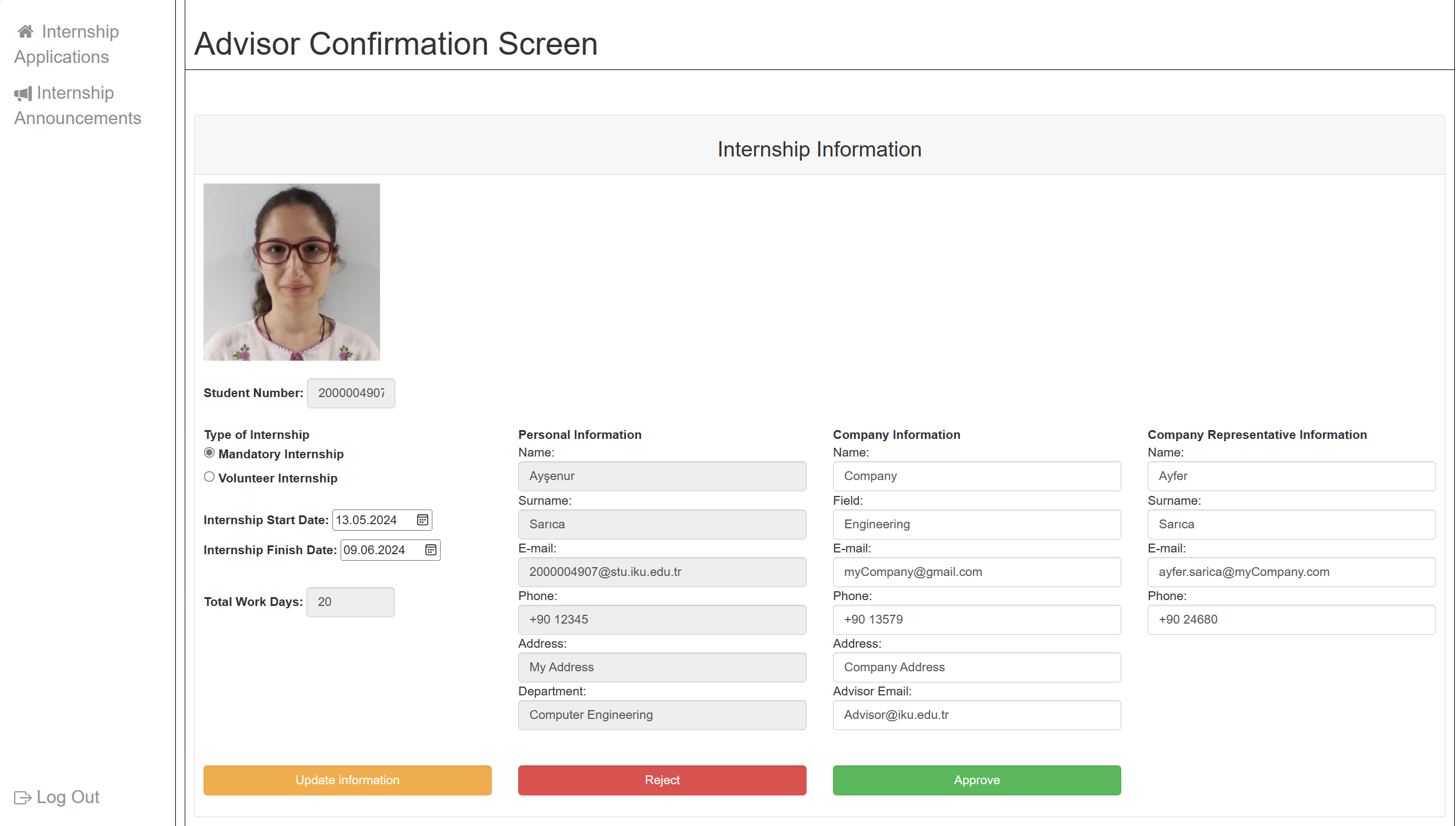


Figure 8: Advisor/InternshipDetails Page Edit

When advisor approved the internship application, its color turns to green. **(figure 9)**

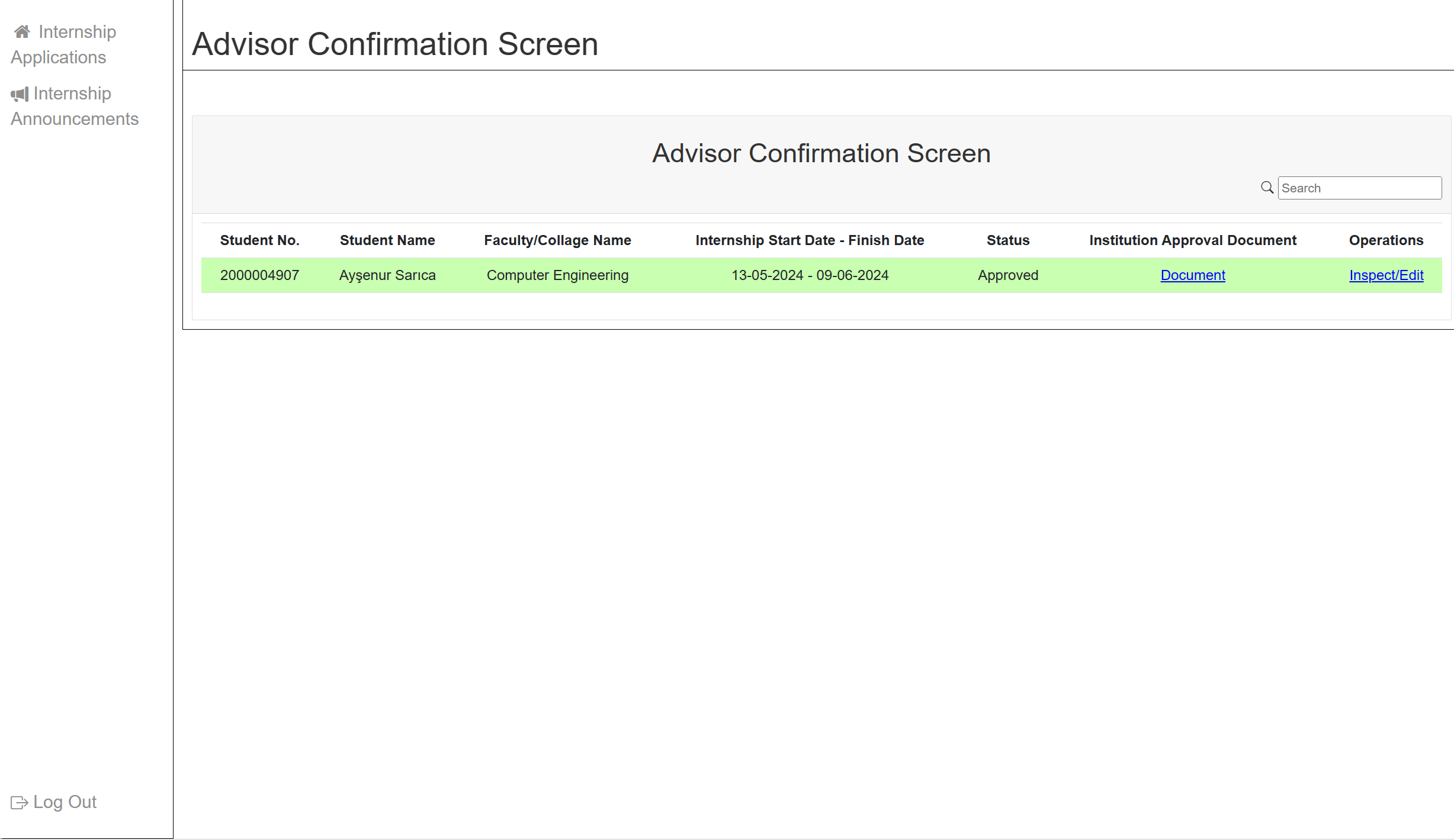


Figure 9: Advisor/index Page

Admin enters the start day, end day and description into the calendar. **(figure 10)**

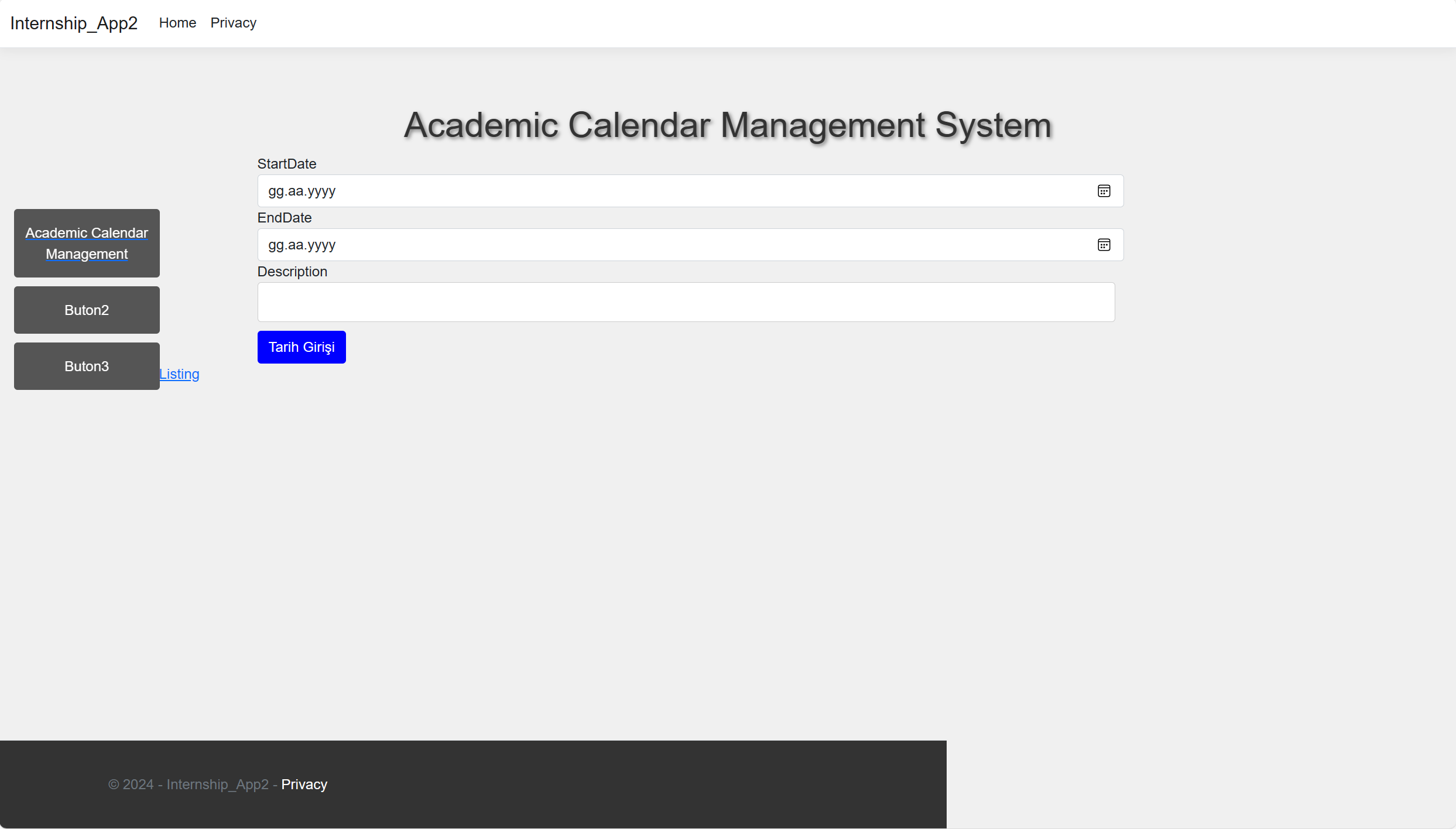


Figure 10:Admin/Calendar Page

# Test

Student number and password were entered on the login page. The page logged in to the site by sending a request to CATS and the student number appeared on the internship page.

After entering the necessary information such as type of internship, internship start date, internship finish date, personal information, company information and company representative information and uploading the necessary files such as profile picture and internship document, the form was successfully submitted. The information has been successfully saved to the database. It was observed that the site did not submit the form by giving a warning when required fields were left blank.

Total work days were calculated correctly when internship start date and internship finish date were selected. If the total work days for the mandatory internship are not 20 or 40, the site gave an error. Such a situation was not observed for the Volunteer internship.

Submitted internship applications appeared as waiting approval on the advisor page. If the Advisor approves or rejects the internship, the status changes to approved or rejected.

##### References

1. M. T. S. Rugupathi, “Learning ASP.NET Core MVC Programming” What is ASP.NET?, pp. 13-14, 2016.
2. D.Sarka, M. Radivojevic, and W. Durkin “SQL Server 2017 Developer’s Guide” Introduction to SQL Server 2017, pp. 8, 2018.
3. J. N. Robbins “Learning Web Design A Beginner’s Guide To HTML, CSS, JavaScript, And Web Graphics” Hypertext Markup Language (HTML), pp. 12, 2012.
4. J. N. Robbins “Learning Web Design A Beginner’s Guide To HTML, CSS, JavaScript, And Web Graphics” Casscading Style Sheets (CSS), pp. 12, 2012.
5. J. N. Robbins “Learning Web Design A Beginner’s Guide To HTML, CSS, JavaScript, And Web Graphics”JavaScript/DOM scripting, pp. 13, 2012.
6. Coursera, C-Sharp, 22 Feburary 2024.