ITRW 324 Project 2: Cyber Security

**Cybertronix**

**Student numbers:**

L.Wassenaar: 25948083

B.Lubbe: 24928410

N.Hefer: 26098903

D.Pieterse: 27740757

**Table of contents:**

[1.1 Software Project Management Plan(SPMP) 1](#_Toc526715201)

[1.1.1 Overview 1](#_Toc526715202)

[1.1.2 Software selection 1](#_Toc526715203)

[1.1.3 Project schedule 2](#_Toc526715204)

[1.1.4 Group member roles and Responsibilities: 2](#_Toc526715205)

[1.1.5 Process model 3](#_Toc526715206)

[1.1.6 Security technical aspects 3](#_Toc526715207)

[1.2 Software requirement specification(SRS) 4](#_Toc526715208)

[1.2.1 Description 4](#_Toc526715209)

[1.2.2 Specific requirements 5](#_Toc526715210)

[1.2.2.1 User interface 5](#_Toc526715211)

[1.2.2.2 Hardware interface 5](#_Toc526715212)

[1.2.2.3 Web application interaction 5](#_Toc526715213)

[1.2.3 Functional requirements 5](#_Toc526715214)

[1.2.4 Performance requirements 6](#_Toc526715215)

[1.2.5 Design Constrains 7](#_Toc526715216)

[1.3 ER Diagram and DFD 7](#_Toc526715217)

[1.4 Web application design 8](#_Toc526715218)

[1.5 Software to install on windows 8](#_Toc526715219)

[1.6 Conclusion 12](#_Toc526715220)

[2.1 References 13](#_Toc526715222)

**List of Tables**

[Table 1: Member roles and responsibilities table 2](#_Toc526715102)

[Table 2 : Functional requirements 5](#_Toc526715103)

[Table 3 : Constraints placed on the design 7](#_Toc526715104)

List of Figures

[Figure 1: Gannt chart of project schedule using MS Project 2](#_Toc526715086)

[Figure 2: Process model of project using draw.io 3](#_Toc526715087)

[Figure 3 : Use case of the leave system 4](#_Toc526715088)

[Figure 4: Project ERD reversed engineered on MySQL workbench 7](#_Toc526715089)

[Figure 5 : Project DFD using draw.io 8](#_Toc526715090)

Project 2

## Software Project Management Plan(SPMP)

### Overview

The aim of Software management planning, is to describe technical aspects to collaborators of the project and users. The project consists of creating an employee-leave web application, where the employees of a company can submit applications for a specific kind of leave. Administrators is responsible for adding a specific kind of leave. The data of the specific type of leave and employees must be stored and protected in a database. To protect the company and its precious data, passwords will be encrypted. The purpose of the system is to notify corresponding managers when a leave application is submitted. The manager can then either accept or deny the application.

Project content, like the user-interface, database and security approaches that we used will be discussed, where each group member was assigned to a specific role. The project workload was distributed over an agreement of roles, where each member had a specific role to successfully complete this project (see figure 1, page 2).

### Software selection

We used PhpStorm as an editor for PHP (using the Laravel framework) and MySQL for the Relational Database.

Main features/benefits of PhpStorm:

* Smart code assistance
* Completion of code and syntax highlighting
* Incorporates the structures of WebStorm.
* Enables editing for HTML and JavaScript.
* User friendly interface
* Refactoring can be automated.
* Includes well-rounded kits for testing and debugging.
* Contains tools for connecting and working with databases.

Main features/benefits of MySQL:

* User friendly interface.
* Protects confidential data by using data safekeeping layers.
* Greatly optimized
* Support for many different development interfaces.
* Designed to manage large amounts of data.

Main features/benefits of Laravel (See 1.1.6)

### Project schedule

Figure 1: Gannt chart of project schedule using MS Project

### Group member roles and Responsibilities:

Table 1: Member roles and responsibilities table

|  |  |  |
| --- | --- | --- |
| **Name:** | **Roles:** | **Responsibility:** |
| Natassja Hefer | Front end programmer | * User interface design planning * Designing the user interface * WebApp documentation |
| Brandon Lubbe | Back end programmer | * Database implementation (physical design) * Development of web application * Database and web-app testing |
| Deon Pieterse | Back end programmer | * Database implementation (physical design) * Development of web application * Database and web-app testing |
| Leonard Wassenaar | Administrator | * Database conceptual design * SPMP documentation * SRS documentation |

### Process model



Figure 2: Process model of project using draw.io

### Security technical aspects

Security is an important aspect of this project, and therefore we decided to use the Laravel framework. Laravel is used as a framework for PHP and has gained an enormous amount of popularity. Laravel takes care of many security issues that a programmer would’ve made provision for. Data security is important in any business sector, because the business depends on this data and it allows the business to operate without interruptions like SQL injections.

Main security features of using the Laravel framework:

* The framework provides an Authentication system by utilizing guards to handle how the authentication of the users will be done for every request.
* SQL injection prevention that ensures passed data aren’t immediately applied in a SQL query.
* Passwords are secured by Bcrypt. This makes it impossible for someone to decrypt the password.
* Cross site request forgery protection that prevents third level parties from making forged requests.

## Software requirement specification(SRS)

### Description

This segment will provide a birds-eye view of the web-application. Three main users are emphasised in the system, namely administrators, managers and employees. Each one of the users has a different motive in the system with their own specific requirements. An administrator can only create a profile for a new employee/manager and create specific leave types. Employees can then apply for these leave types created by the administrator. The manager should only then be able to view leave applications and accept or reject the applications from employees.

Figure 3 : Use case of the leave system

### Specific requirements

#### User interface

A new user will see the login form as soon as the user opens the web-application. The users should then be able to log in as an employee, manager or administrator. An administrator has the ability to create a new account on the web-page. Administrator should have a form page for adding new leave types and register or update a manager or employee. Managers should be able to view pending applications. Employees must have a form page to apply for a specific kind of leave.

#### Hardware interface

Since this is a web application, an online connection is a necessity, and therefore hardware is required that can connect to the internet. Examples of media to be used, Modem, wi-fi and LAN cables.

#### Web application interaction

The interaction between the database and web-application is mainly GRUD operations, where data can be viewed and modified.

### Functional requirements

Table 2 : Functional requirements

|  |  |  |
| --- | --- | --- |
| **IOP** | **Functional Requirements** | **Depended on** |
| Input: | Administrator registering | None |
|  | Administrator adding employee/manager information. | Admin login |
|  | Administrator updating employee/manager | Admin login and newly added employee/manager |
|  | Administrator adding a new leave type. | Admin login |
|  | Administrator updating a leave type. | Admin login and newly added leave type |
|  | Employee applying for leave | Employee login |
|  | Manager approving/rejecting applications. | Manager login and employee application |
| Processing: | Calculating system statistics. | Number of Employee/managers/administrators registered.  Leave applications quantity |
|  | Calculation of leave days | Leave application |
| Output: | Employees viewing application history. | Employee login and inserted employee applications |
|  | Managers viewing leave applications of employees. | Manager login and inserted leave applications |
|  | Administrators viewing leave types. | Admin login and inserted types of leave |
|  | Administrators viewing employee/manager information. | Admin login and inserted employees/managers |

### Performance requirements

* The system must have quick response time.
* The system must handle large amount of workload, maintaining performance.
* The system must be scalable and adaptable for future needs.
* Database must support large amounts of data.
* Normalising database tables.

### Design Constrains

The following is software design constrains affected by hardware

Table 3 : Constraints placed on the design

|  |  |
| --- | --- |
| Server: | * Bandwidth of 20MiBps to handle approximately 1000 users per second * Primary storage(Ram) of 8Gb * Storage: 2 Tb solid state drive |
| Client: | * Internet connection |

## ER Diagram and DFD

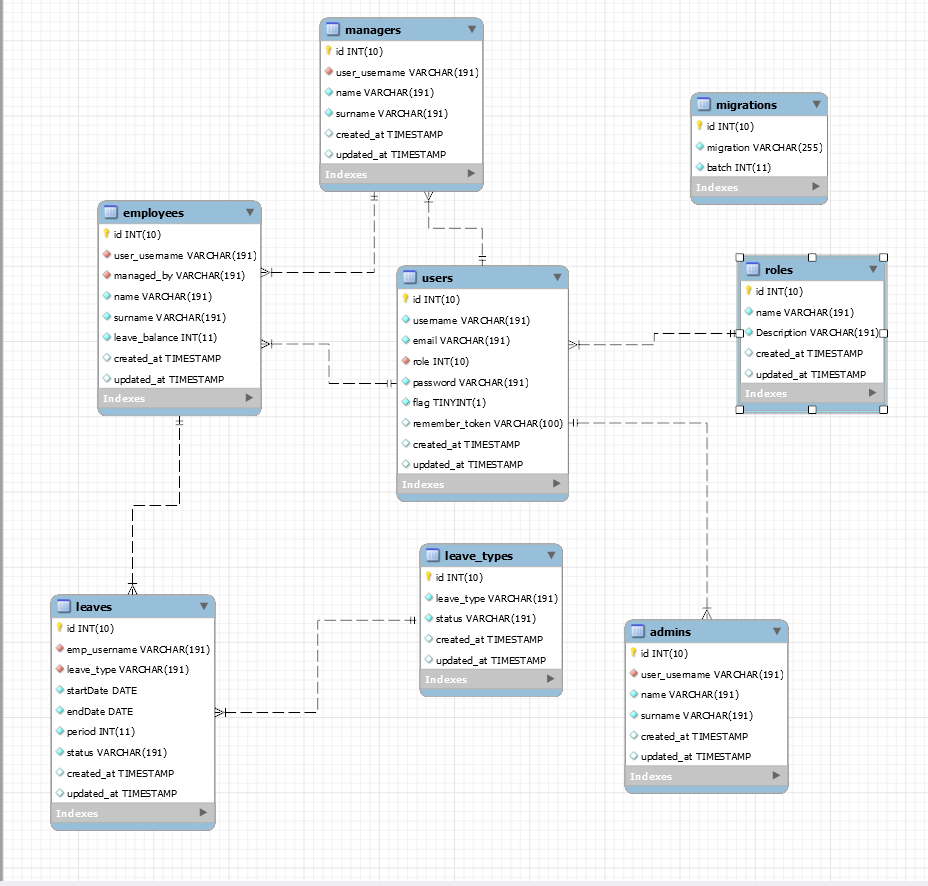


Figure 4: Project ERD reversed engineered on MySQL workbench



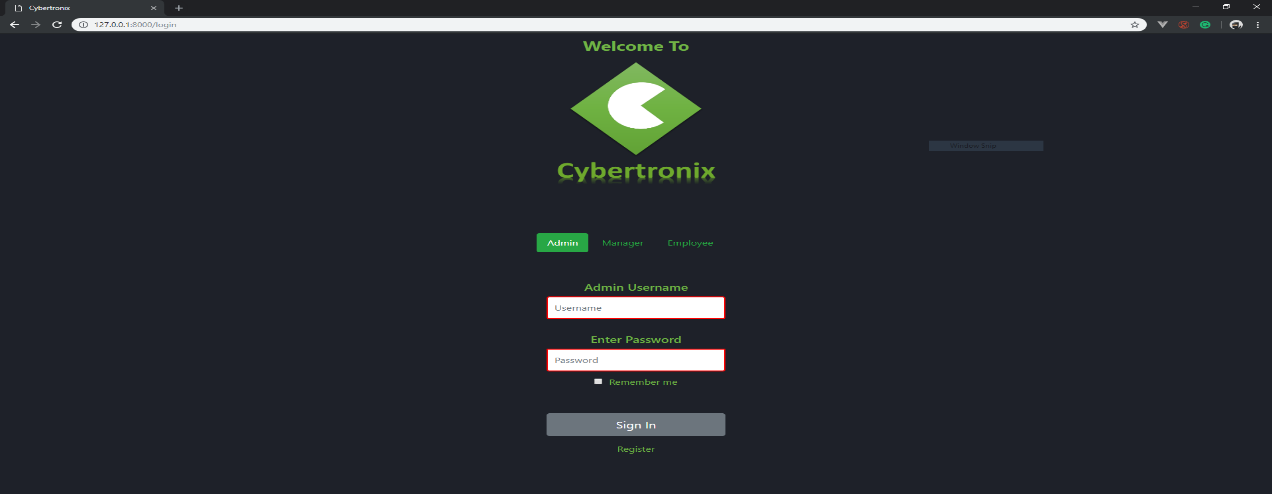
Figure 5 : Project DFD using draw.io

## Web application design

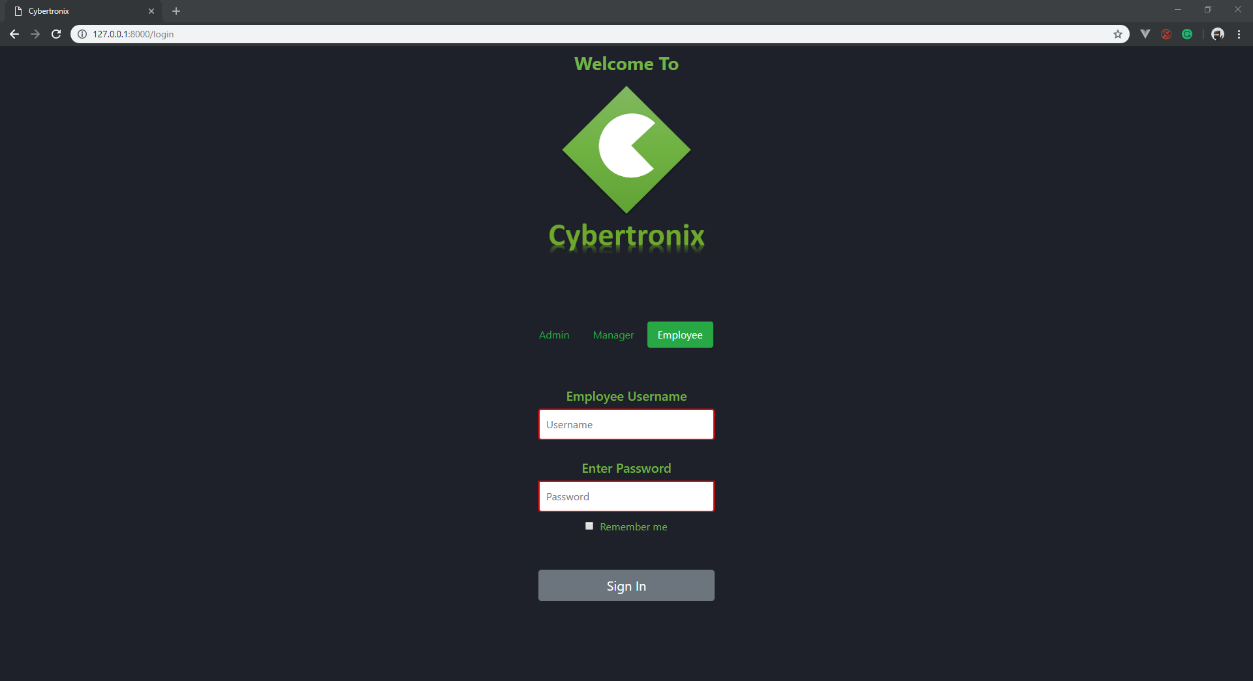
Interface design

Login

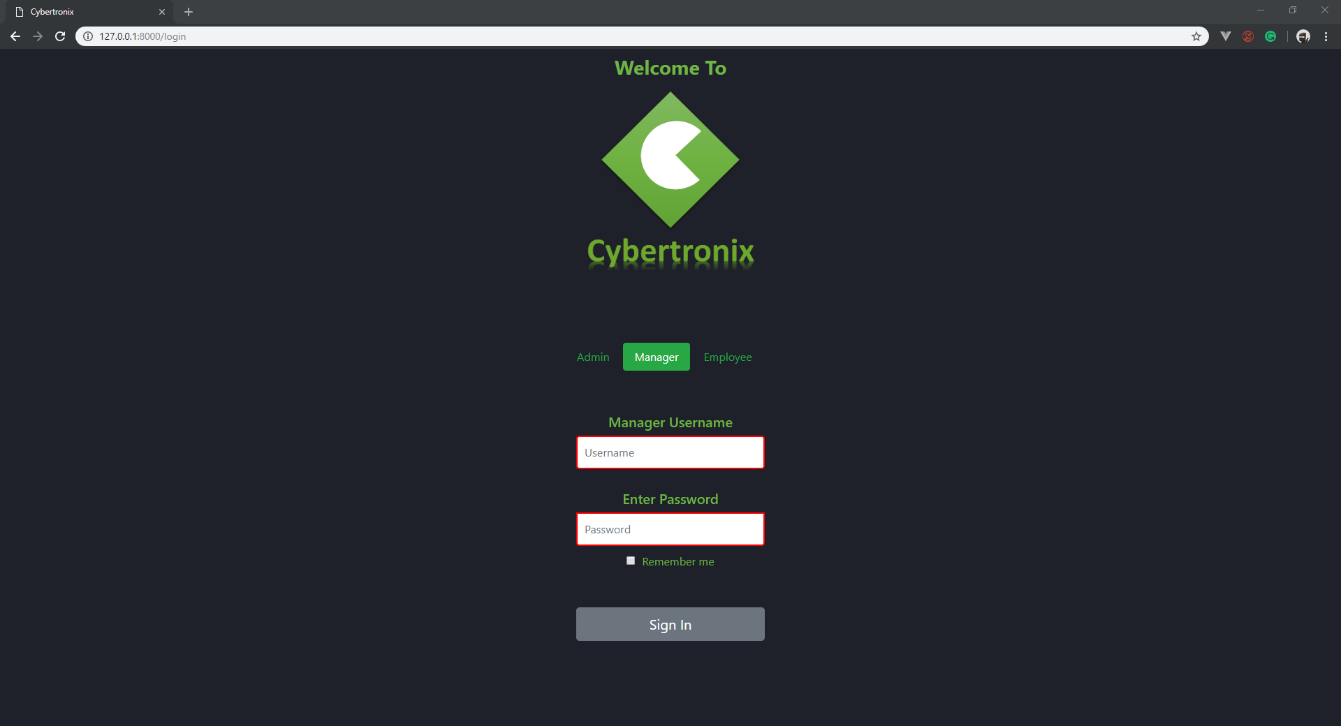
Admin login:

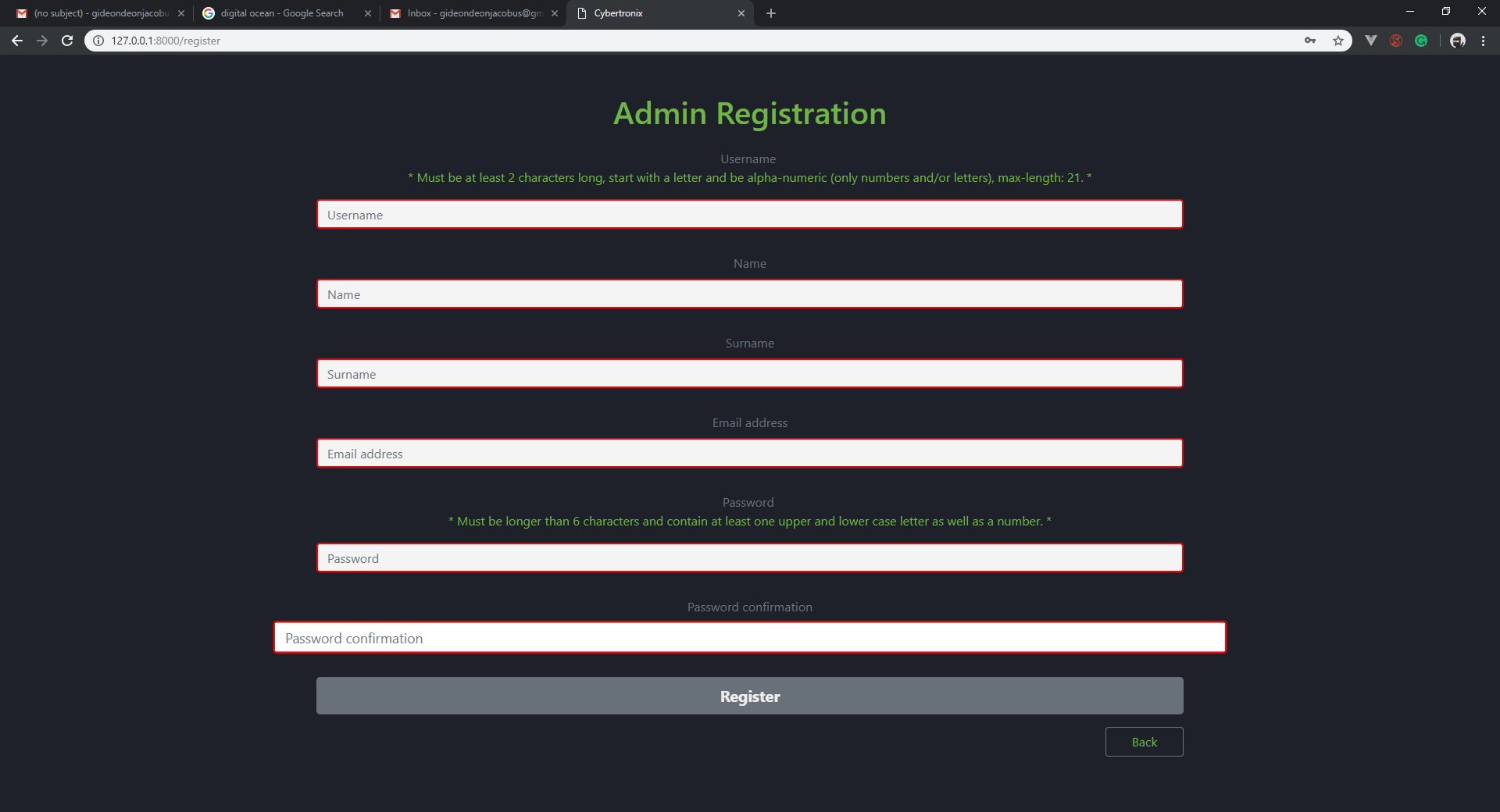


Employee Login:

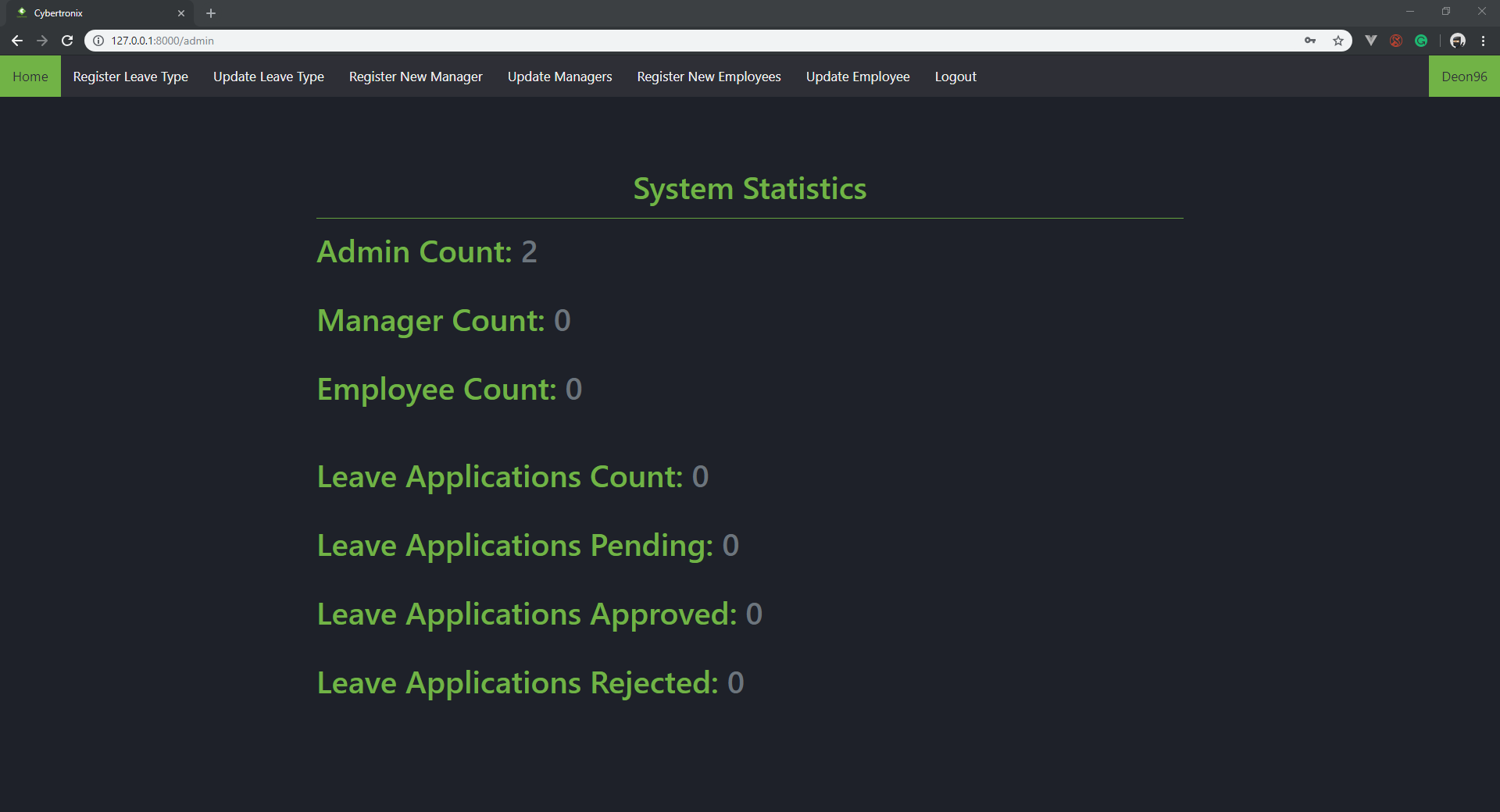


Manager login:

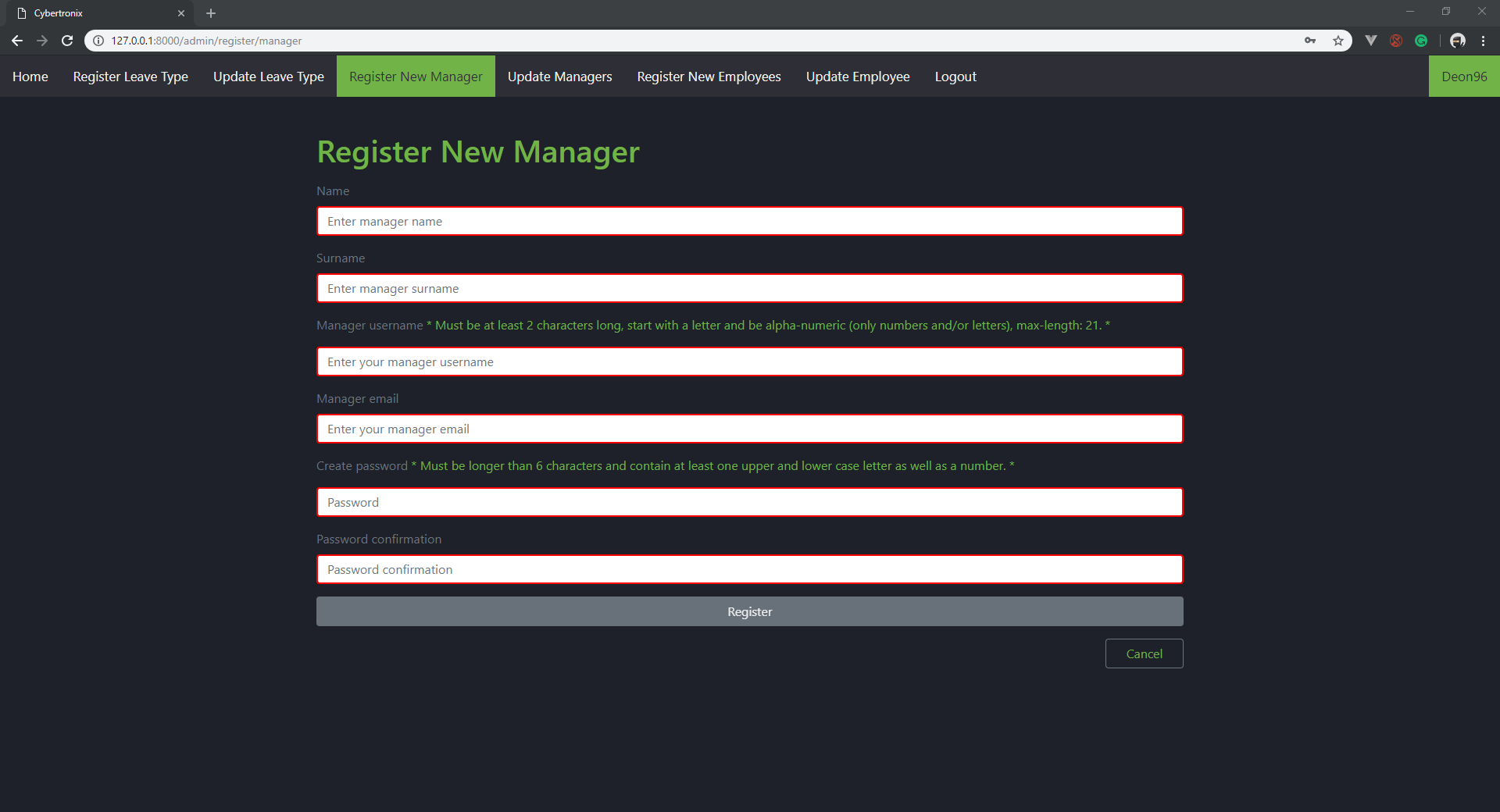


Admin registration page:

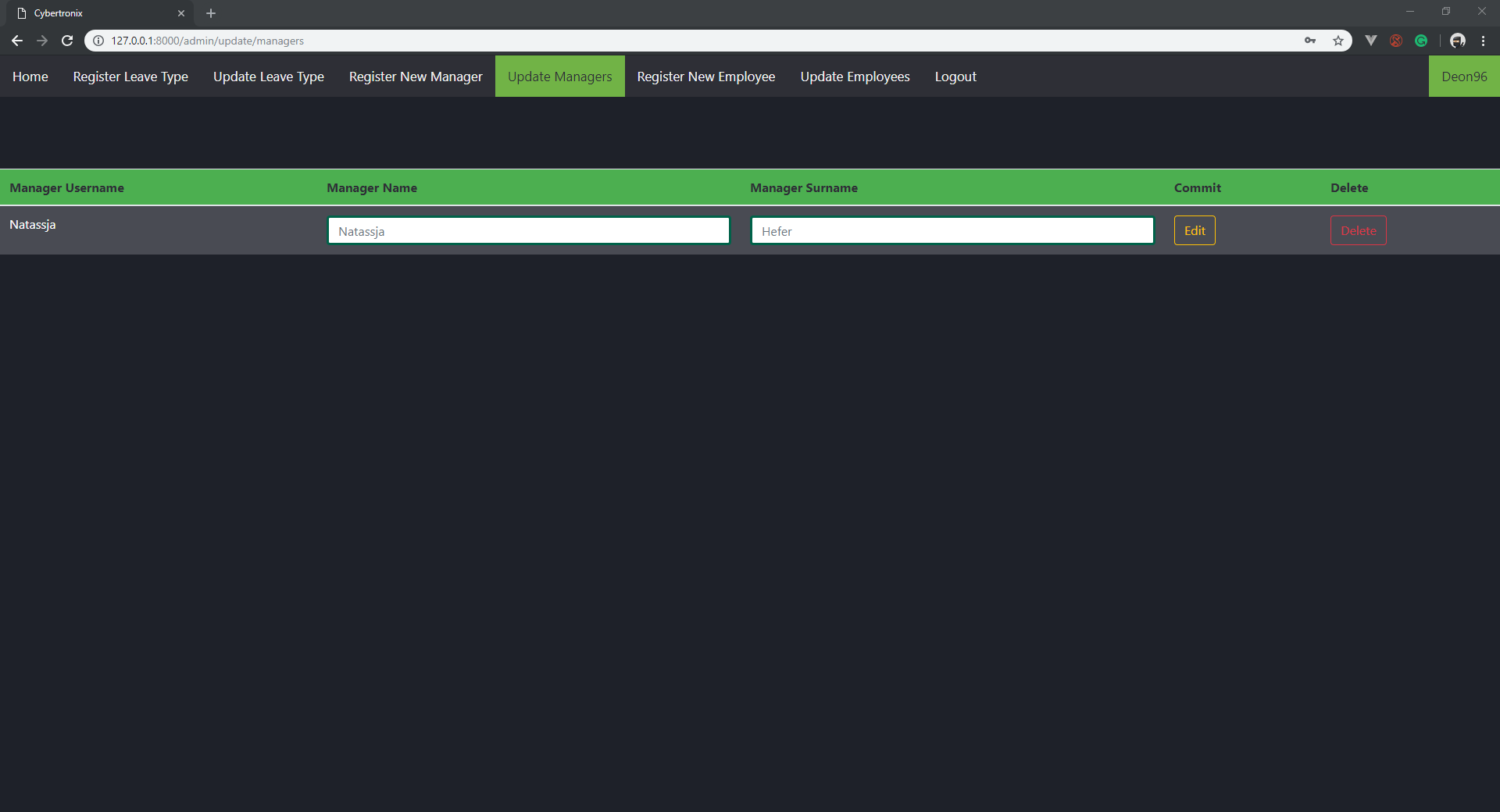
Admin home page:



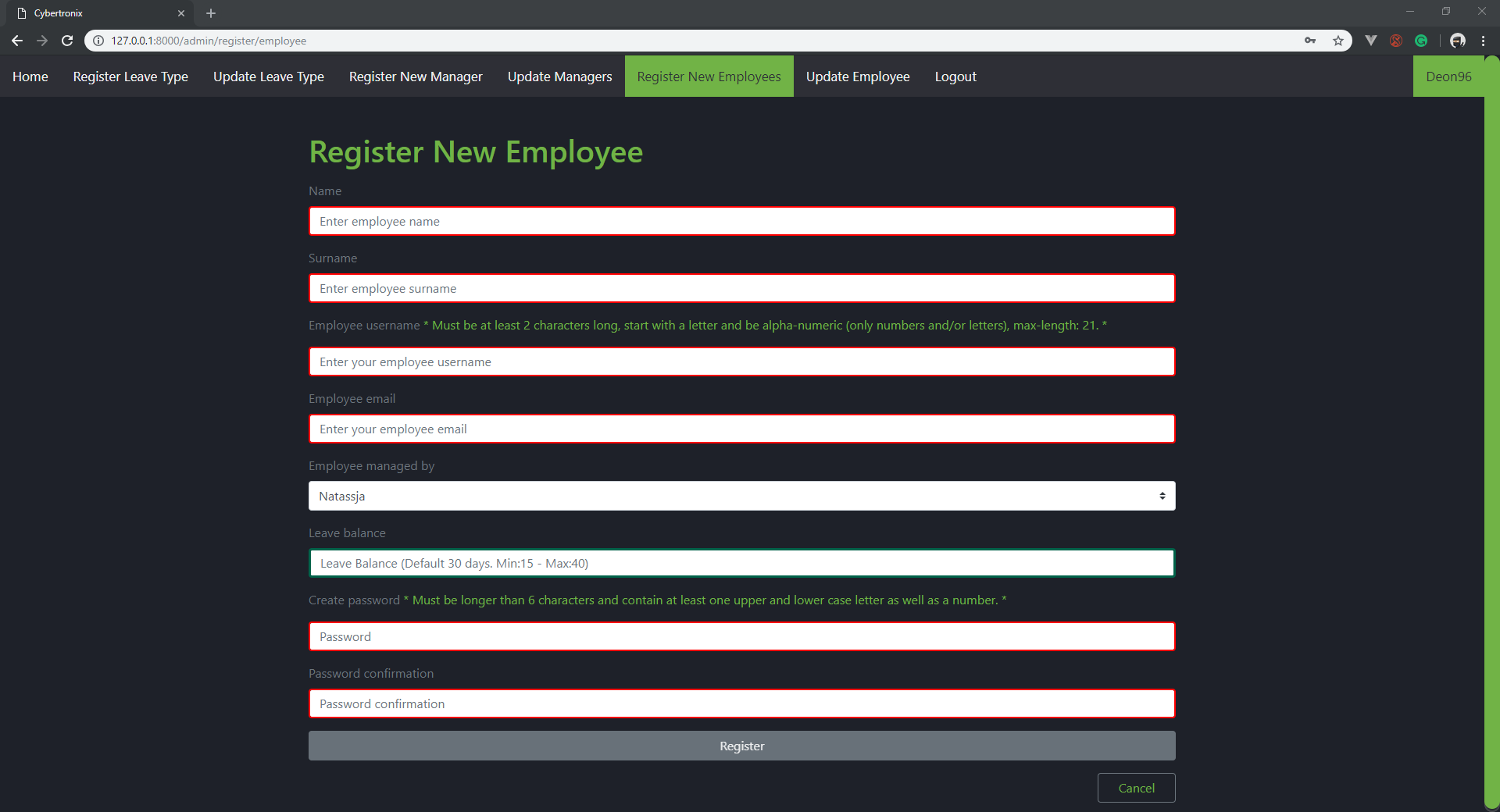
Admin register manager:

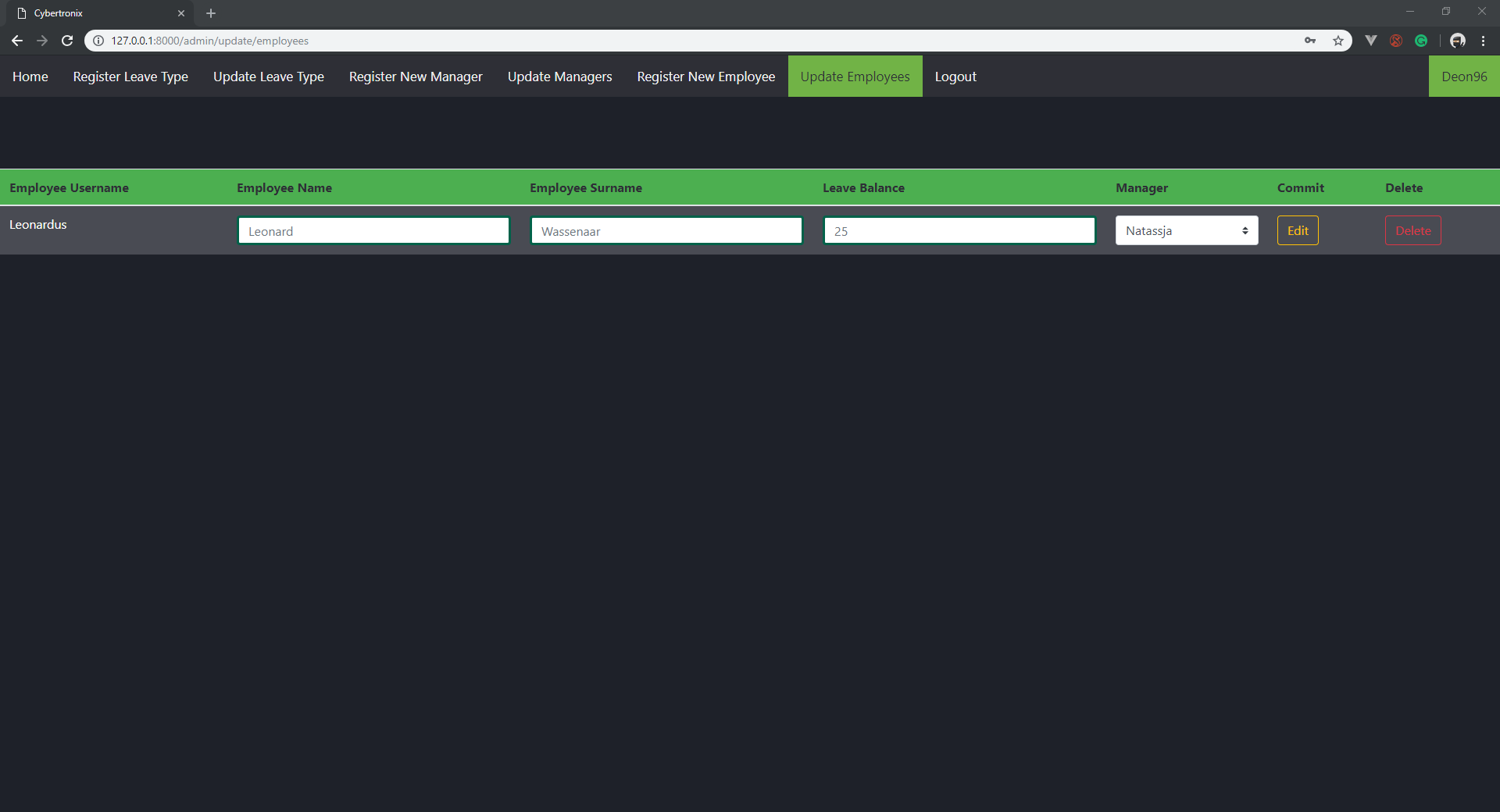


Admin update manager:

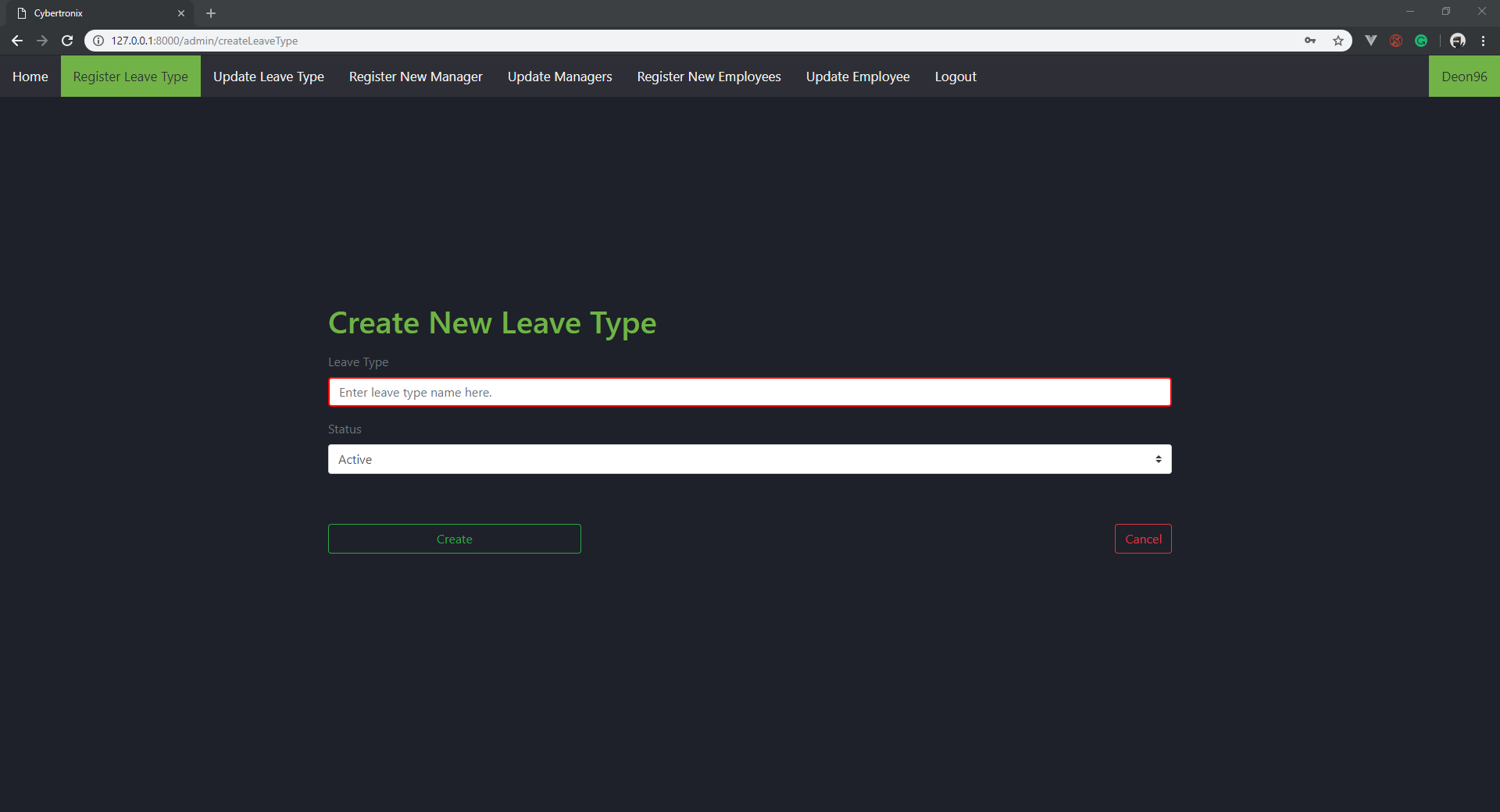


Admin register employee:

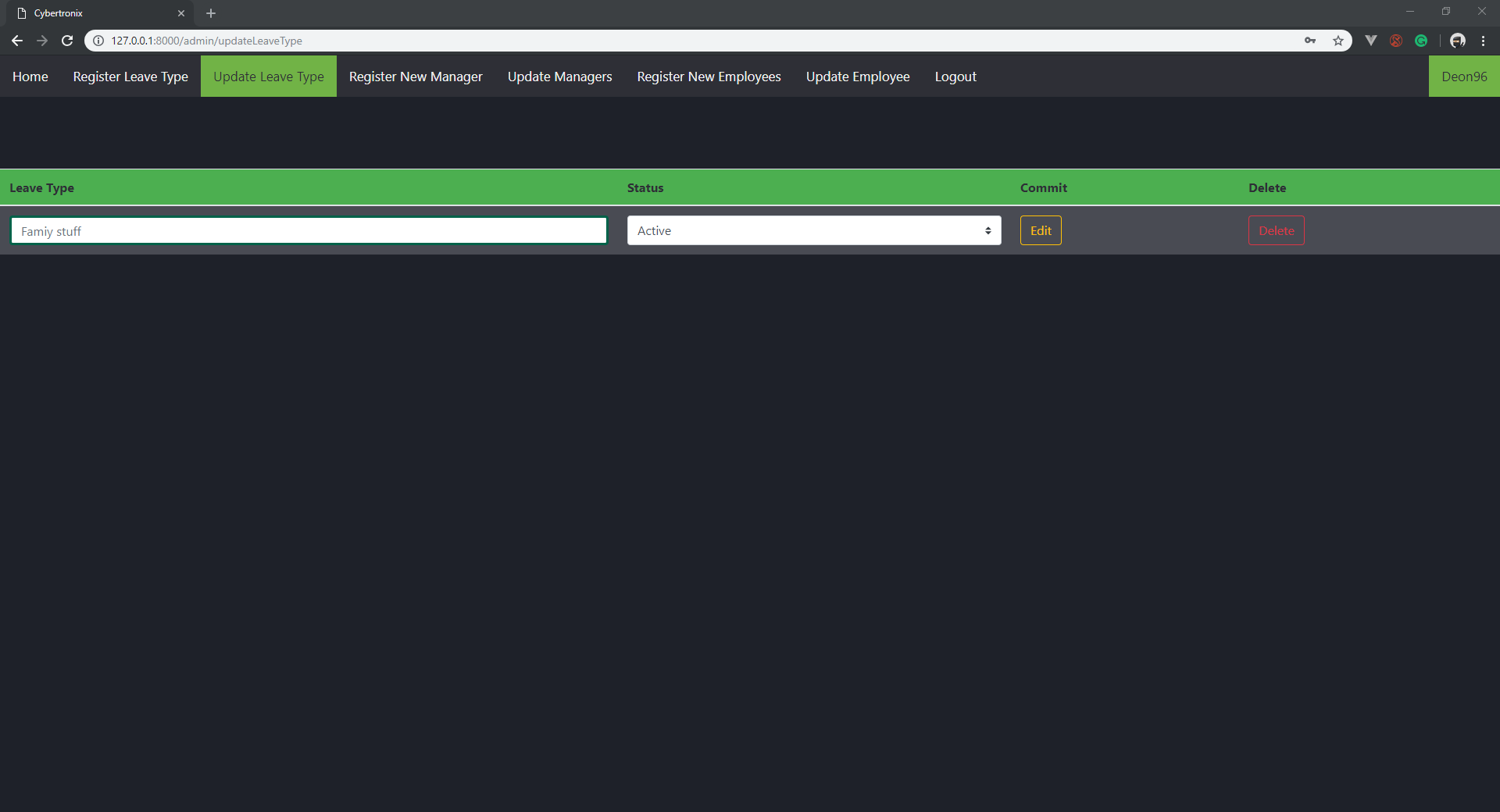


Admin update employee:

Admin create new leave type:

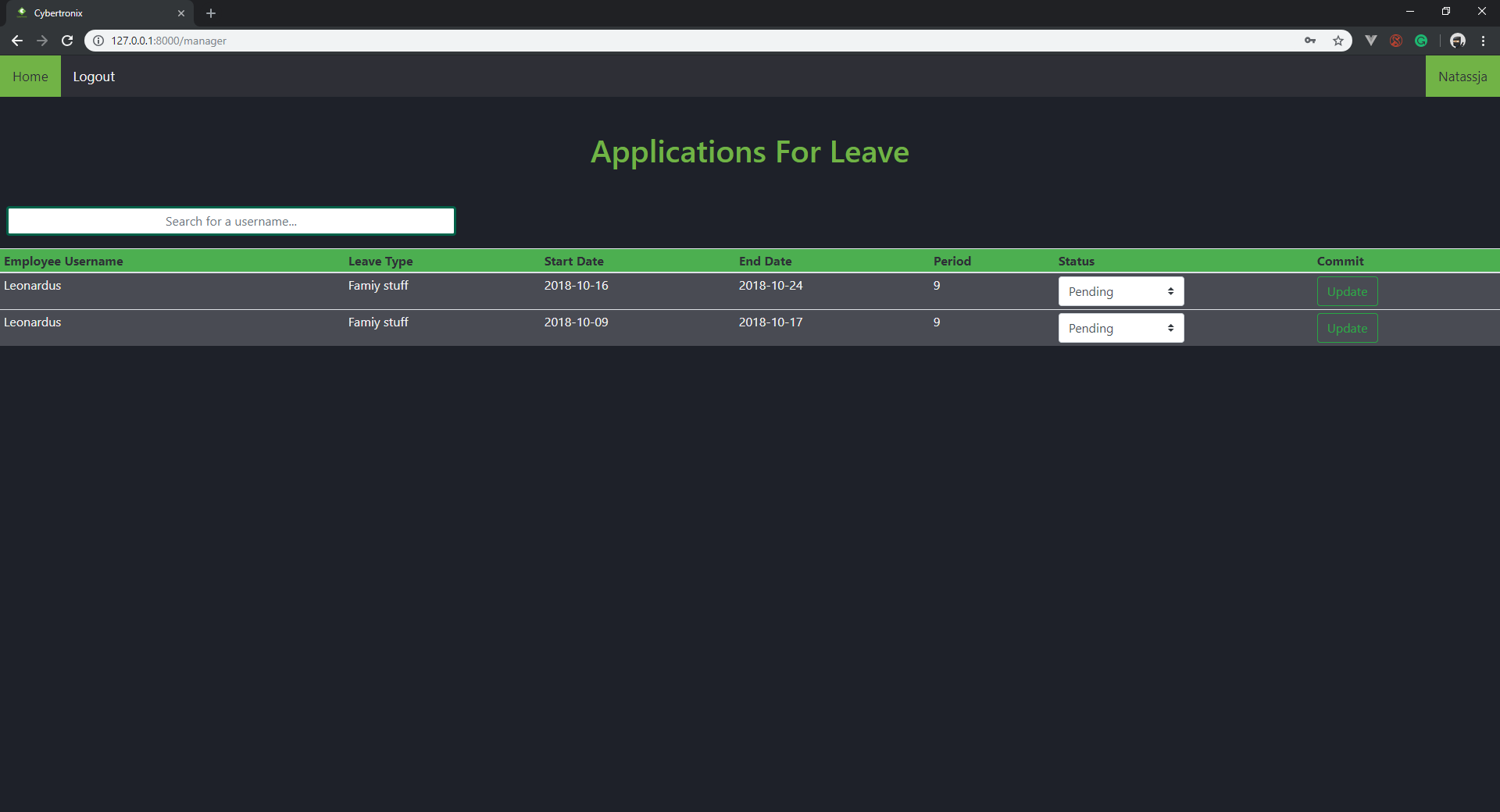


Admin update leave type:

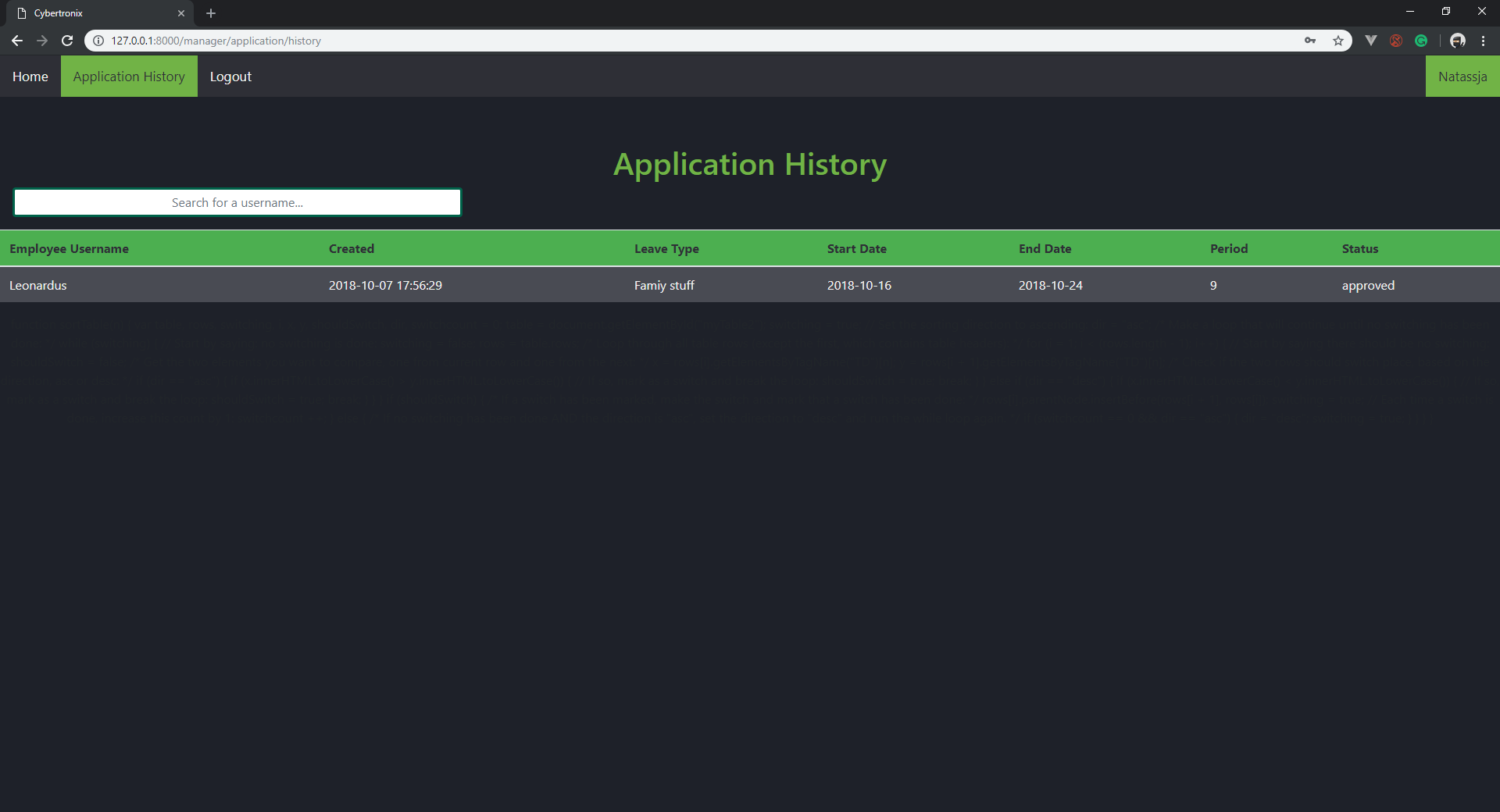


Manager

Manager home page:

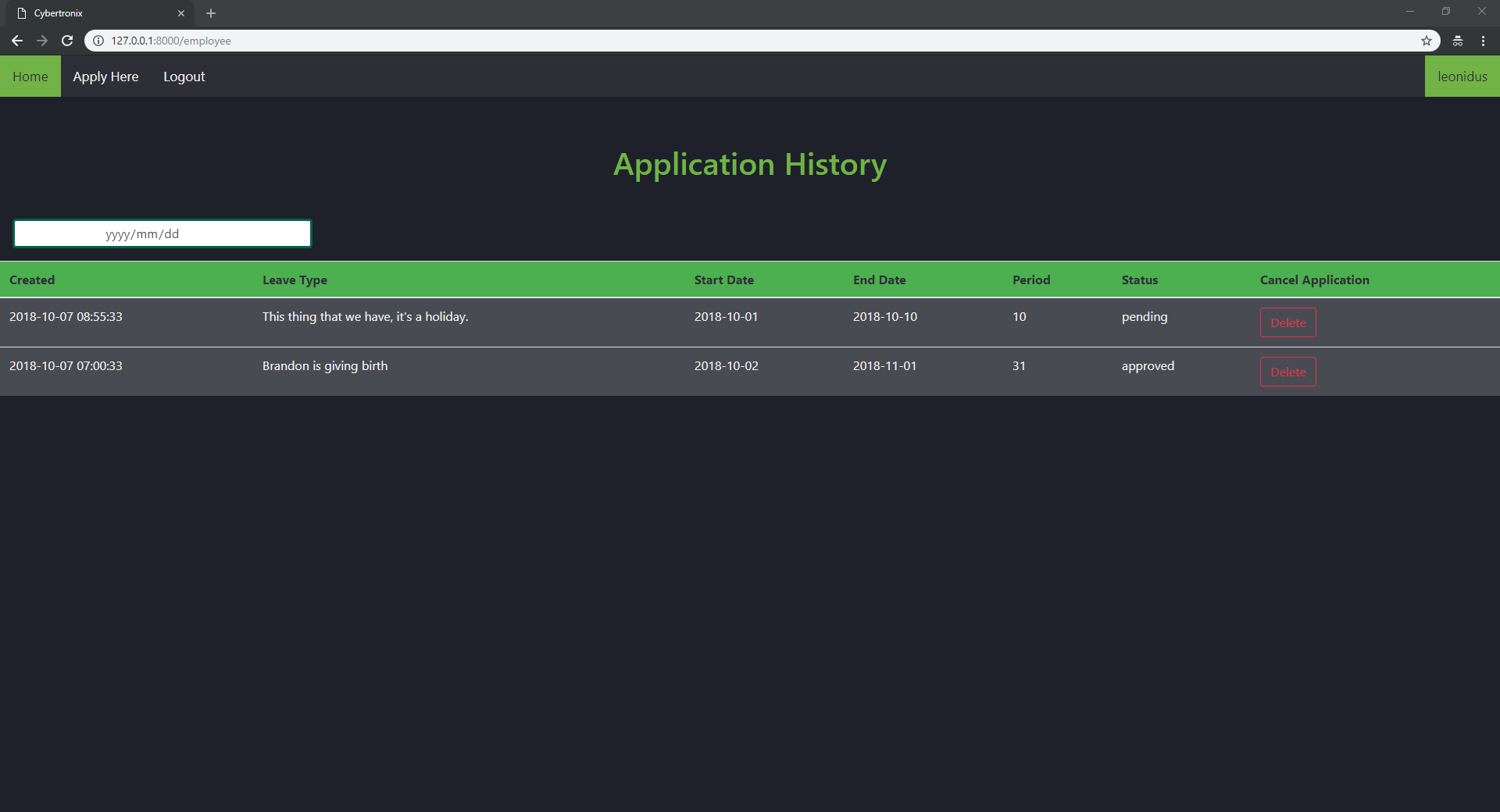


Manager view employee history:

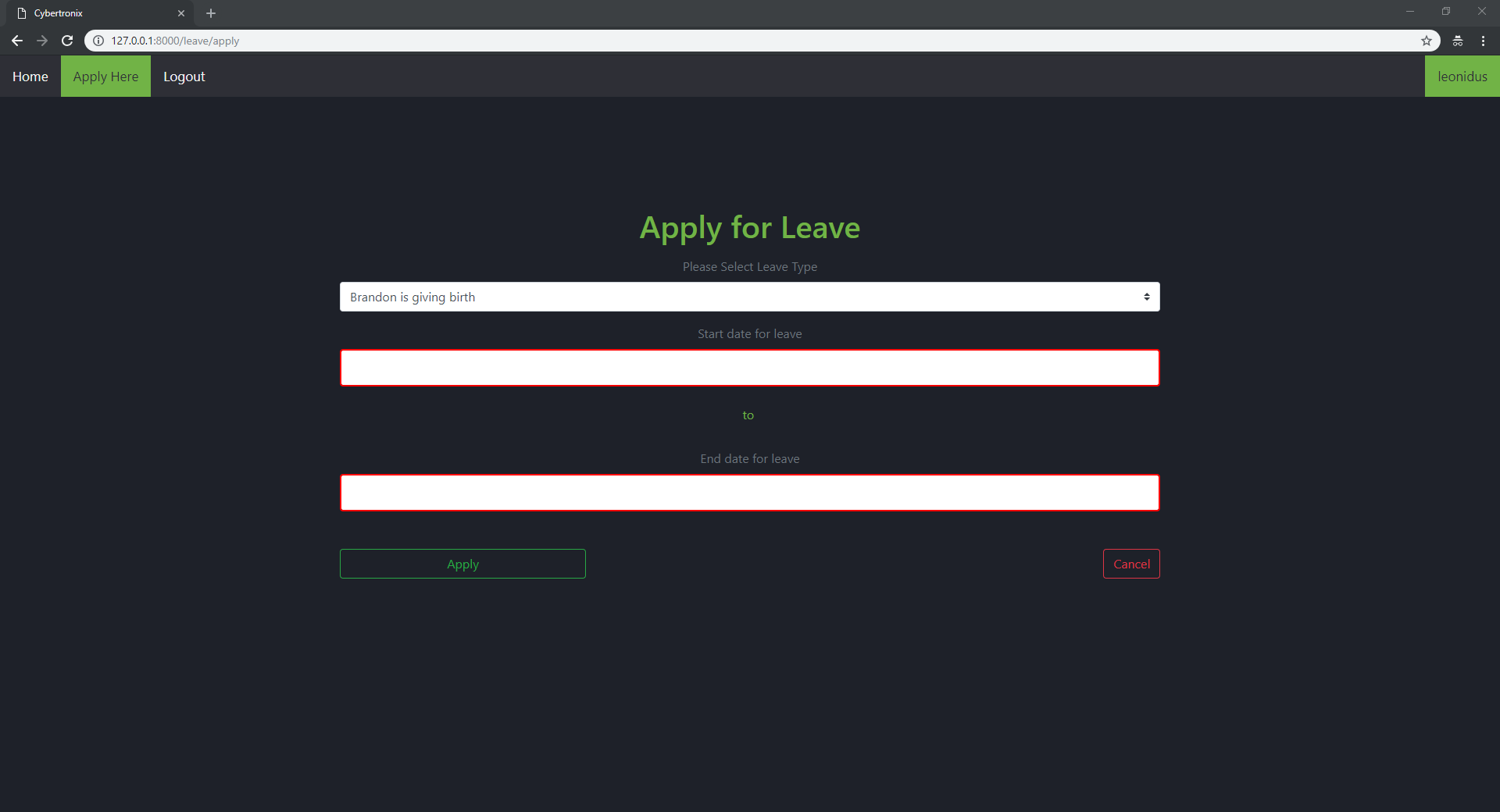


Employee

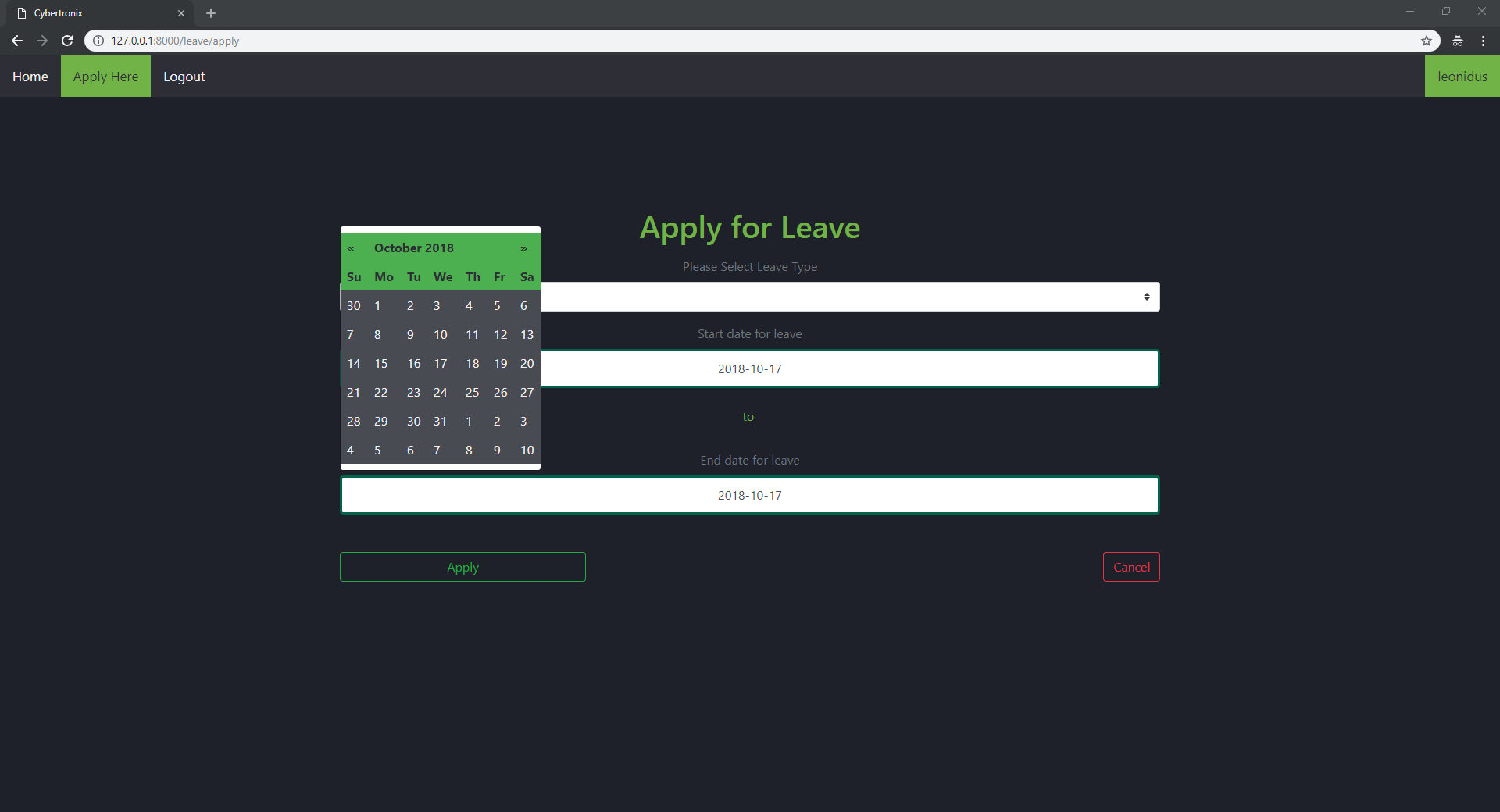
Employee home page:



Employee Apply for leave:

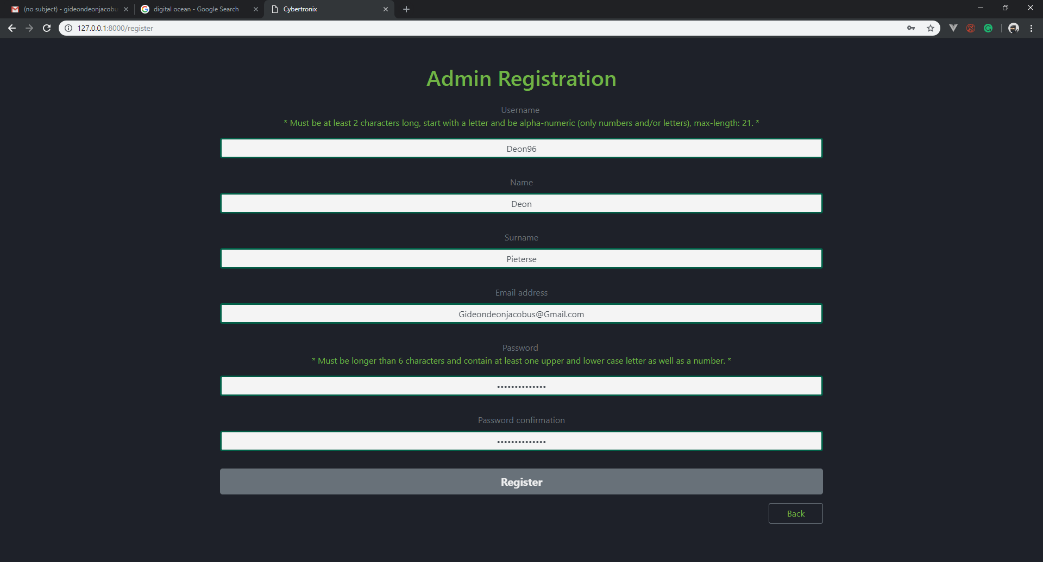


Employee application calendar:



Validation

Example of validation:



Aesthetic design

According to David and Janson (2008) a website’s first impressions is met within the first view seconds. These first impressions often lead to the decision to leave or stay on the website, one factor that influences a user’s decision is the aesthetics of the web page.

The aesthetic design in the web application is strong in the sense that it is a clean and easy to use design that is both attractive and professional to the user. A dark theme with the color green to complement the darkness of the theme is used and gives an appealing look. Consistent colors for validation to avoid confusion for the user is being used for example red for fields that are needed to be filled and green for fields that are filled in correctly.

The darker back ground and the darker green we used for the writing is softer on the eyes of the user which helps when users have to spend long hours looking at the screen. The application’s features were kept to the necessary functions needed for the completeness of this web application to avoid unnecessary clutter and confusion on the web site for the user.

Content design

As mentioned by [Richards (2016](#_ENREF_2)) the content design is usually followed by the following question: “what is the best way to fulfil a user’s needs?”. Which implies that the content design is linked to the user requirements for a web application.

The user needs for this web application is as follows:

Administrator:

Should be able to create employees (CRUD).

Be able to create new leave types (CRUD).

Manager and Employees:

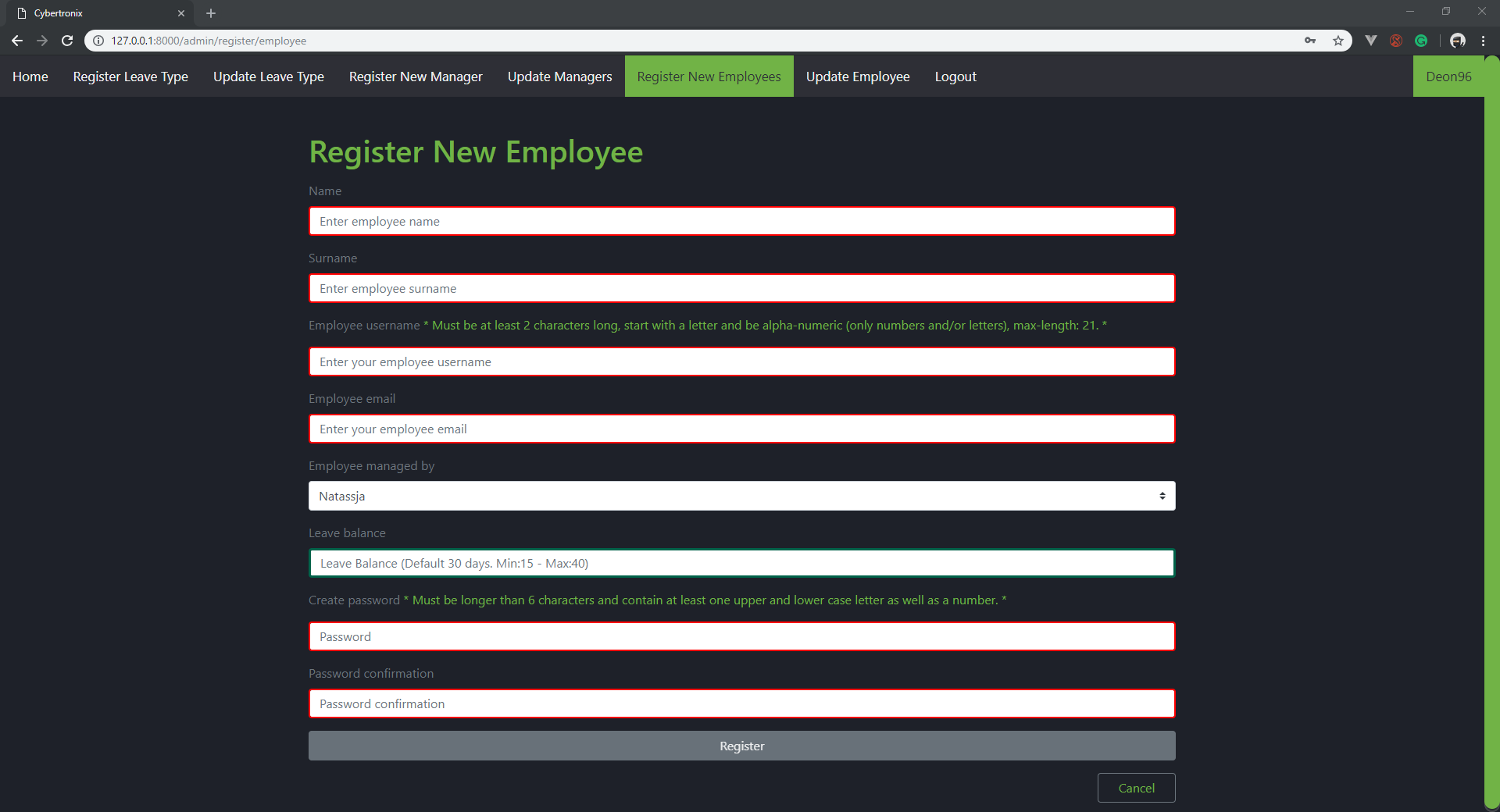
Should be able to view their own leave balances.

Apply for specific leave types with a date range.

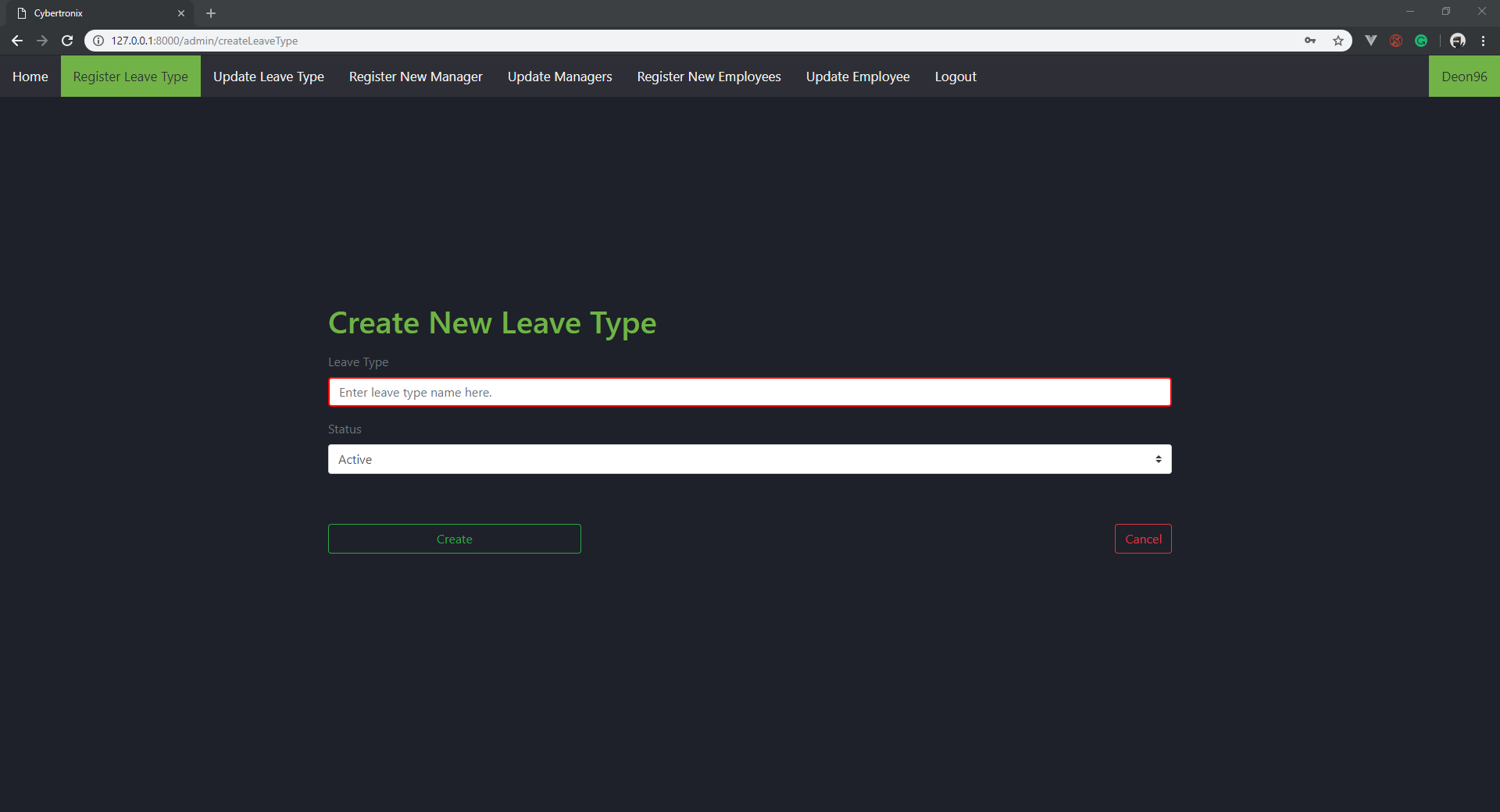
View pending leave approvals.

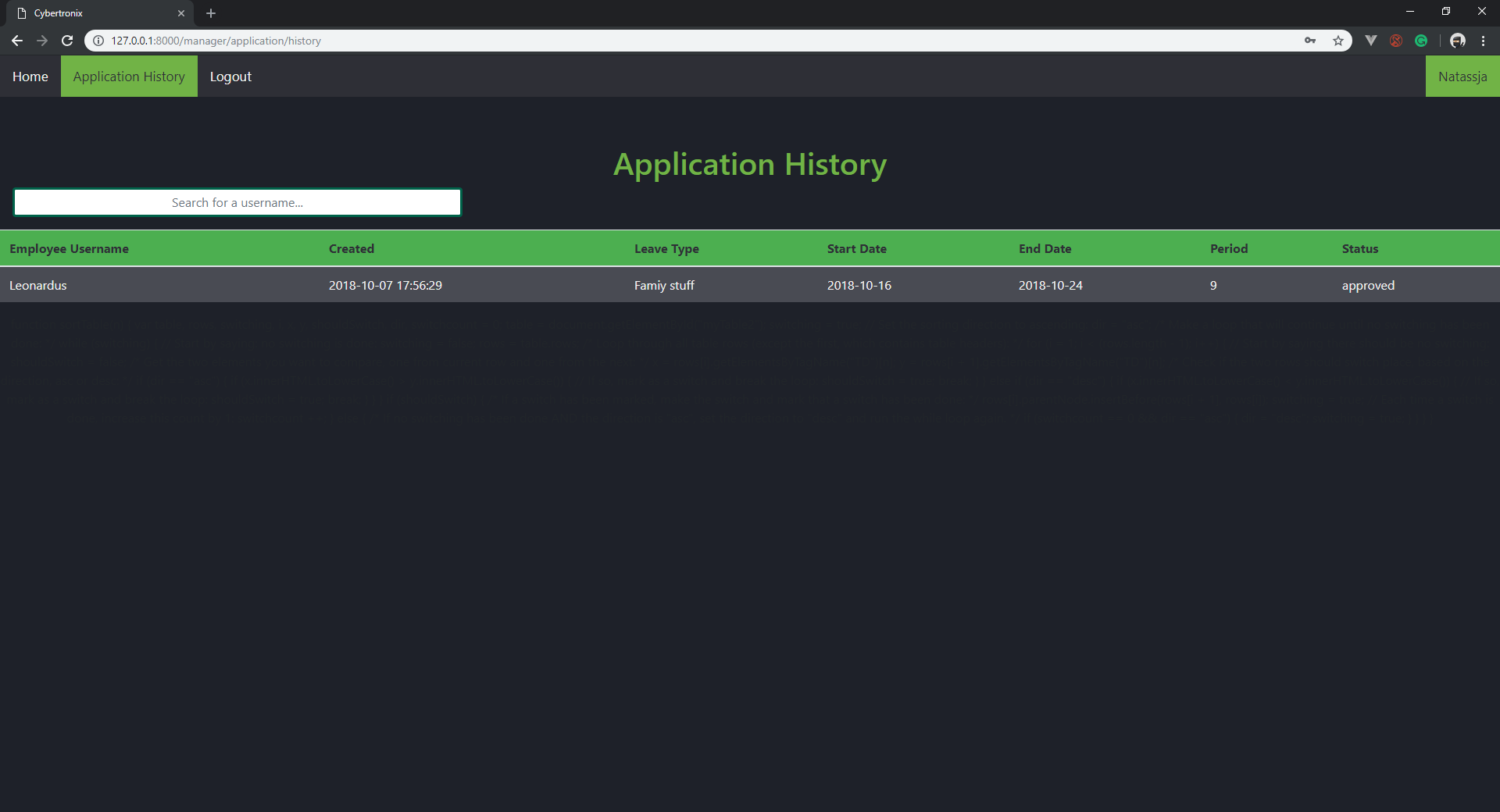
The following is screenshots with explanations on how these requirements mentioned above is implemented in this web application:

Admin creating a new employee. This application makes use of validation text boxes to receive input from the admin and saves the information entered to the database to add the new employee to the system. The admins can use the navigation bar at the top to navigate between form depending on what he/she wants to do.

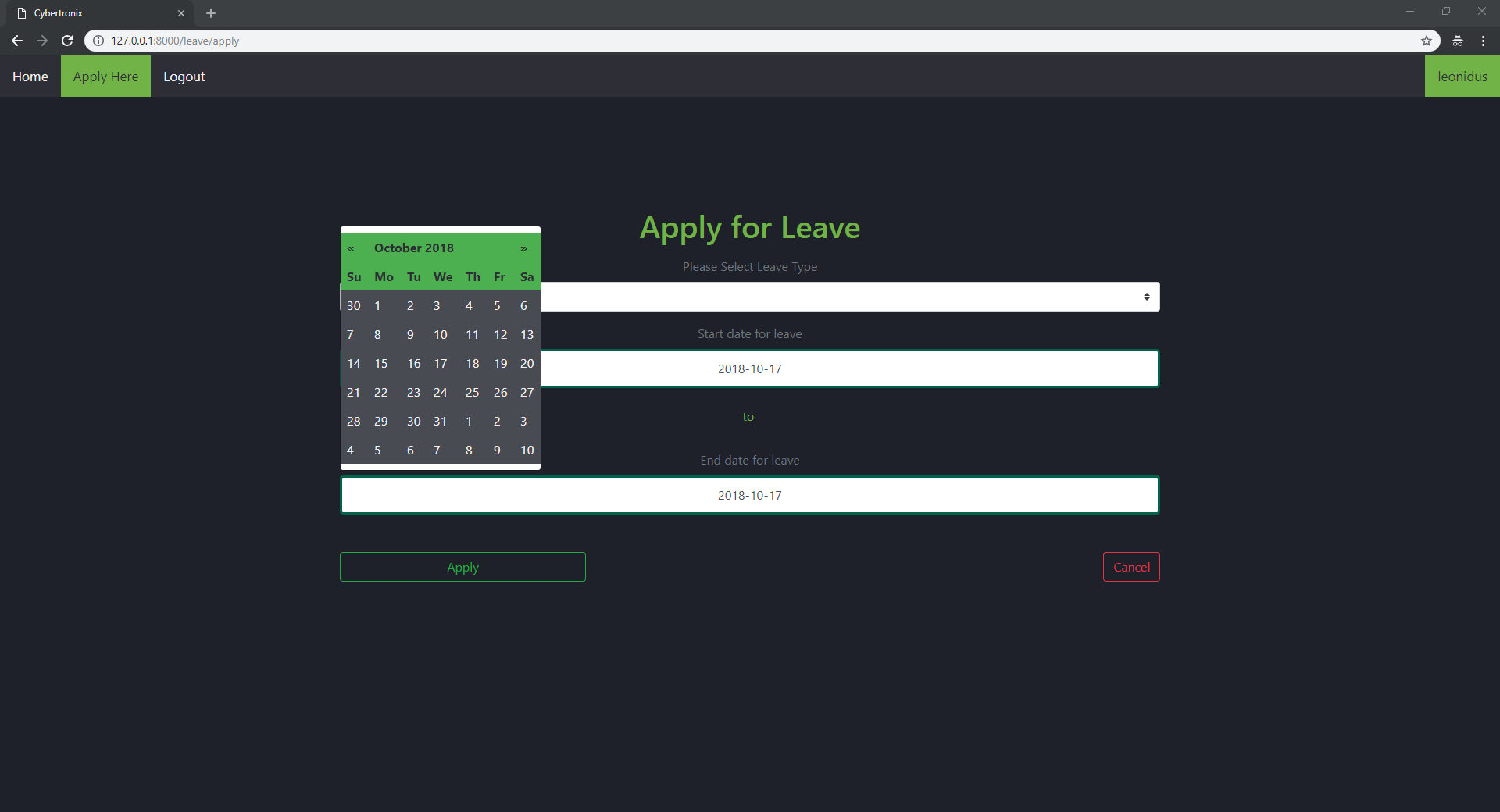


Admin creating new leave types. This application again makes use of validation textboxes to receive input from the admin for detail regarding the new leave type. The information is captured by the create button and stored in the database.

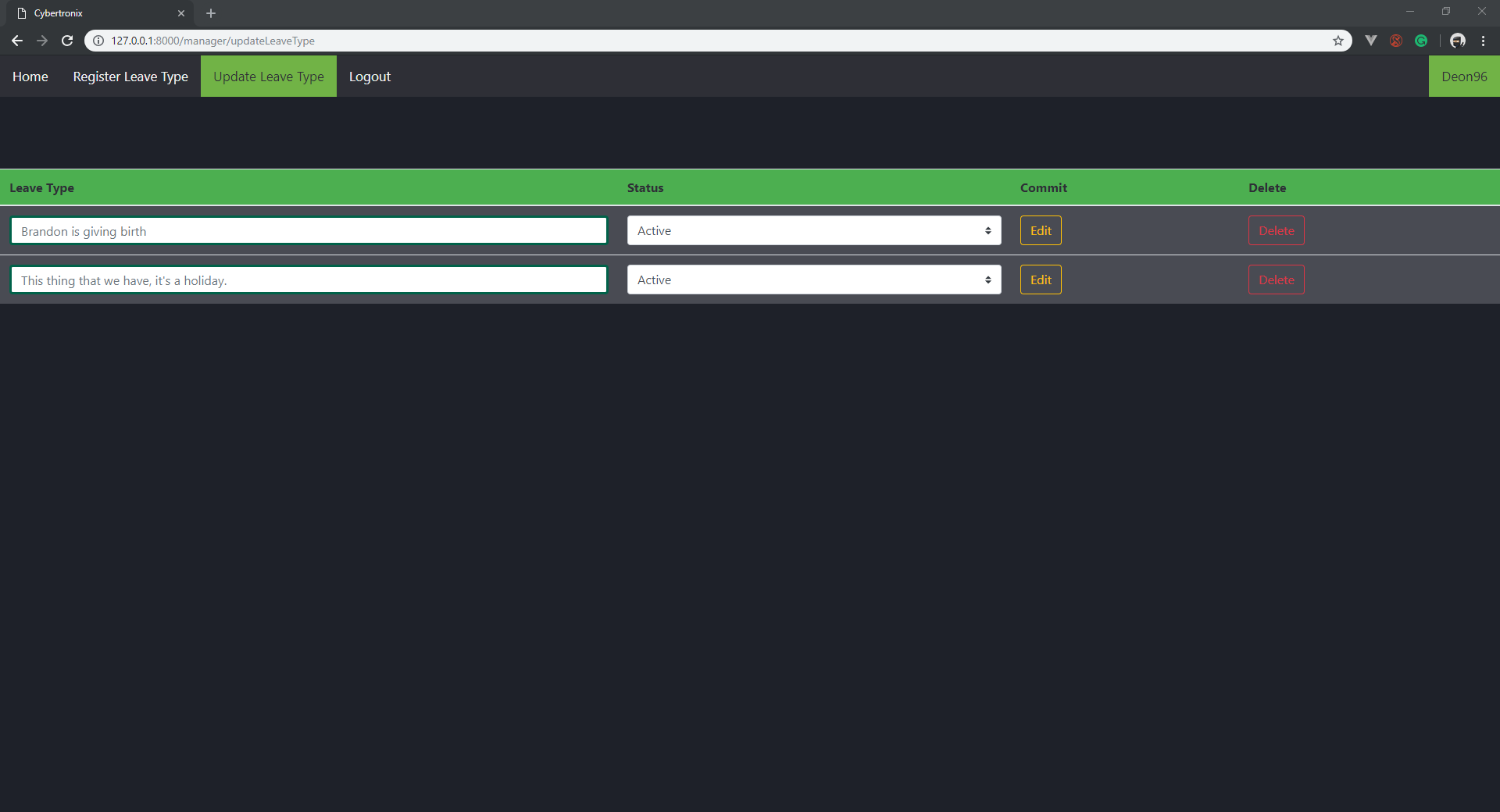


Managers and employees should be able to view their own leave balances. The application uses a table to display the history of applications and weather they have been approved or not. This example is the manager side the employee side can be seen above in the interface design.

The managers and employees should be able to apply for a leave type and provide the range of date they want to take leave. Here we can see the calendar the application uses to capture the date range for the employees to choose when they want the leave. As well as the text boxes to capture details regarding the application for the leave.



Managers should be able to view pending applications to be approved. Here the application makes use of a table to display the pending applications and a drop down list to approve or disapprove the application.



More forms can be seen at the interface design above that helps the application divide the requirements for the different employees to perform their individual roles.

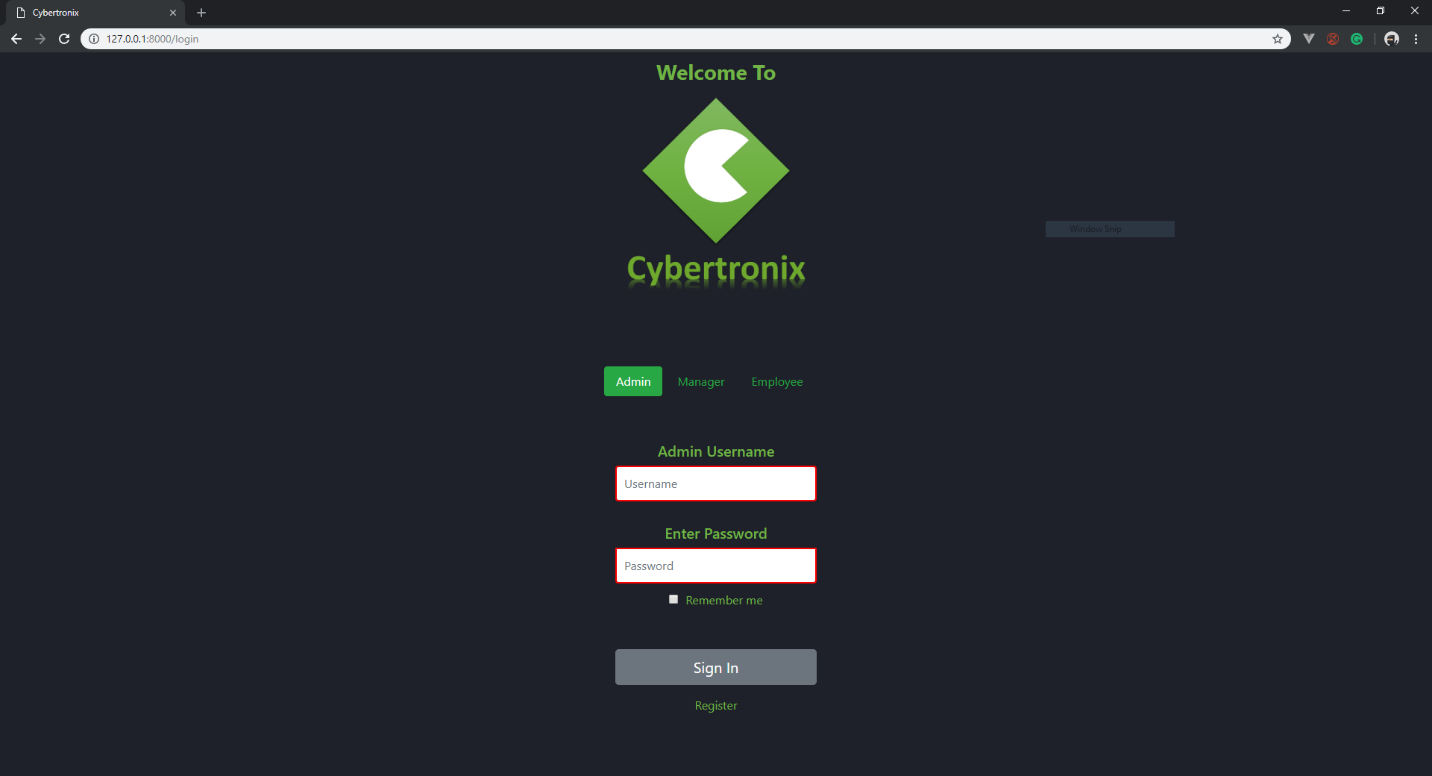
Navigation design component-level design

According to [Anon (2018](#_ENREF_1)) the navigation of a web application should be simple to use and be consistent. The end-user should be able to understand how to navigate from page to page without the need to search for instructions how to navigate through the application.

The following are examples of this web applications navigation components:

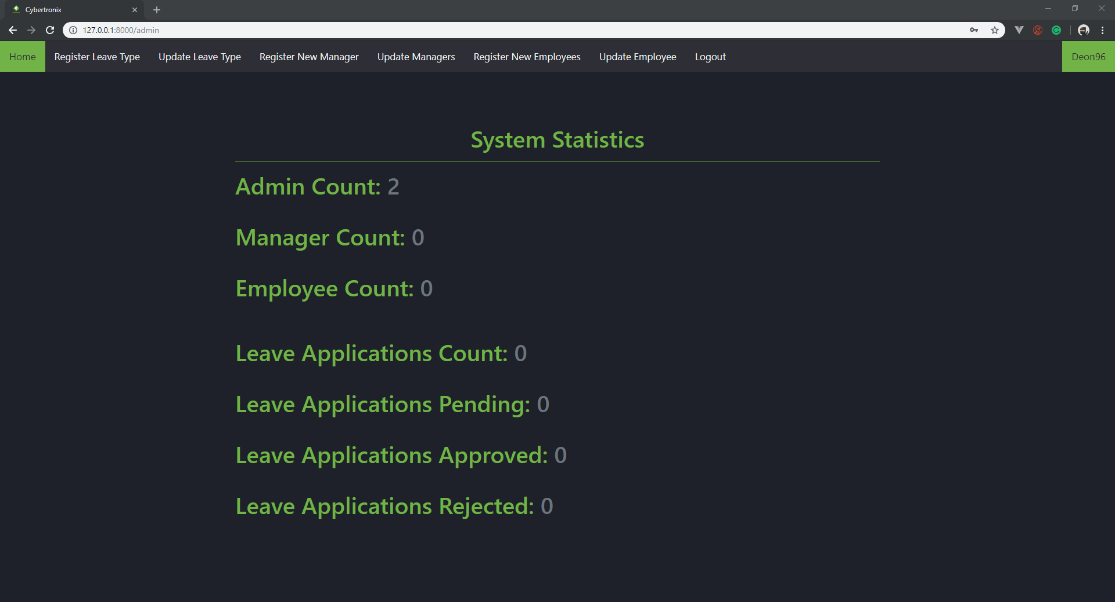
In this figure the login screen can be seen. The user must choose the button whether they are a admin, manager or employee to be able to login with their specific details. If a admin chooses the manager tab they won’t be allowed to login due to their admin details.

Once the details have been entered the sign in button can be clicked to proceed. If a user is not on the system yet they can be added by clicking the register link that will navigate them to the register page.



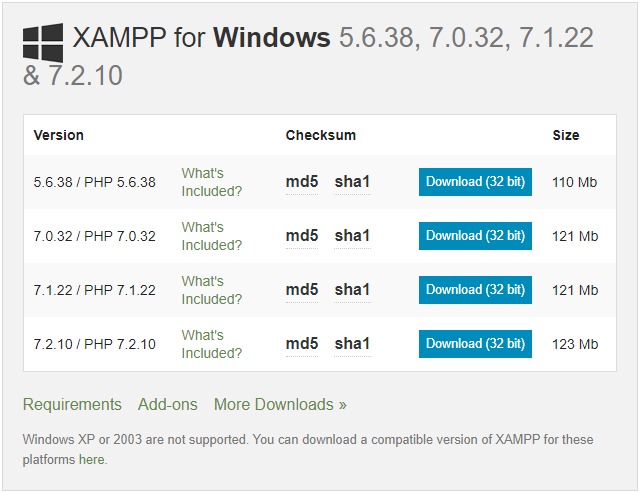
All the forms use a navigation bar at the top of the screen to navigate between pages. Depending on the employee type is what the end user will see in their navigation bar and depending what the user wants to do depends on what tab in the navigation bar they will click.

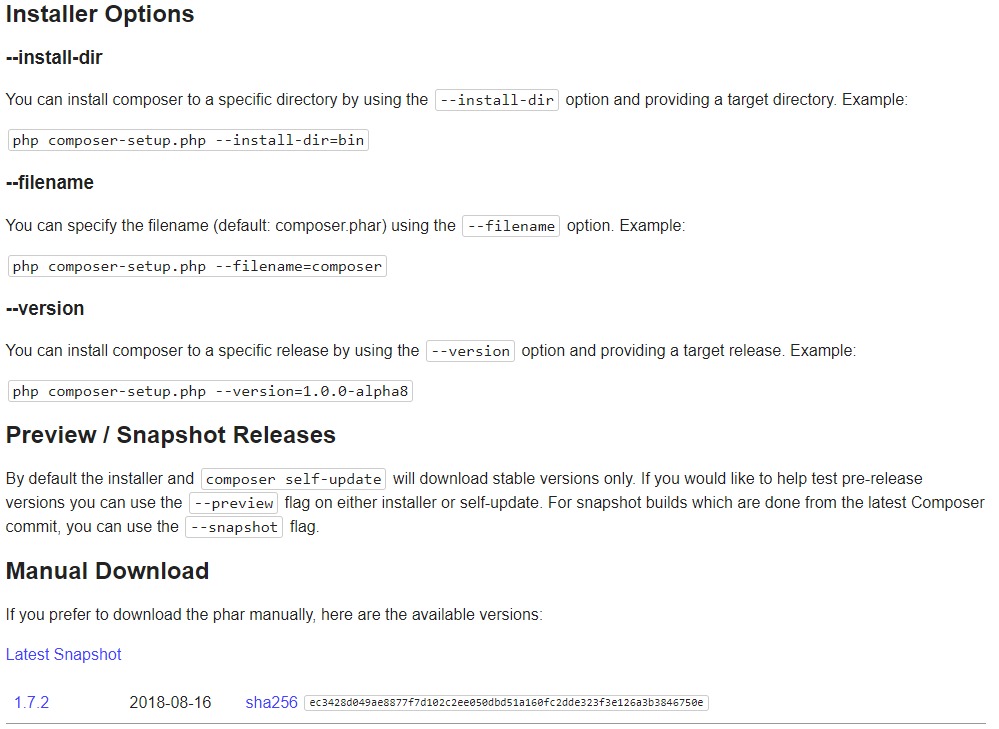
In this example the admin clicks the home tab in the navigation bar then he/she s redirected to the home page for the admins. Each form also has a logout tab in the navigation bar that will link the user back to the login screen of the application.



## Software installation on windows

1. *XAMPP is required for server hosting:*



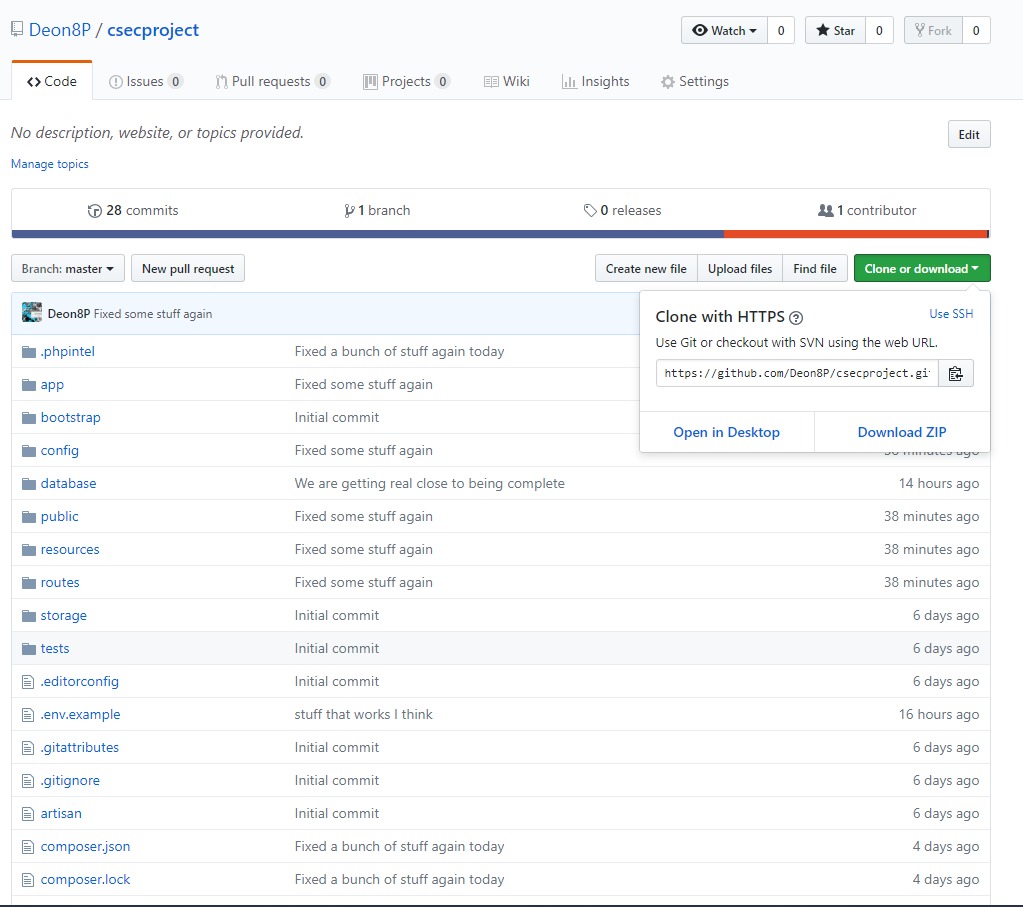
1. *Download composer with one of the following methods*
2. *Install the Laravel framework*
3. *Change directory in gitbash to where you want the repository cloned*



1. *Clone repository (link:* [*https://github.com/Deon8P/csecproject.git*](https://github.com/Deon8P/csecproject.git)*)*



1. *Or the project can be downloaded at* [*http://github.com/Deon8P/csecproject*](http://github.com/Deon8P/csecproject)



1. *Change directory to the project*



1. *Use this command inside the project root directory to host the php server*



1. *View original Laravel files if wanted*

Link: <https://github.com/laravel/laravel>

1. *Nesbot/Carbon packagist if wanted*

(Link: <https://packagist.org/packages/nesbot/carbon>)

1. *Command to require dependency if wanted*



## Conclusion

All the group members have actively worked with their predetermined roles to complete this project. All the group members with their individual roles have noticed one main aspect of the project that can’t be ignored, namely the security of a company’s data. It is more than just storing the company’s data in a database. It is the safeguarding of data to enable a company to continue with their operations. An attack on a company’s data, is a direct attack on the company. By applying all the cyber security approaches, whether they are advanced or basic, will minimize the external threads the company face and ensure smooth business operations.

# Heading 1 won’t print. Don’t delete – doing so will lead to incorrect numbering.

## References

Bently, L & Whitten, J. (2007). System Analysis & Design for. New York, NY: McGraw-Hill/Irwin, pp 246-250.

Branson, T. 2016. 8 Major Advantages of Using MySQL. <https://www.datamation.com/storage/8-major-advantages-of-using-mysql.html>. Date of access: 28 Aug. 2018.

Cameron, D. 2017. Security Features of Laravel 5|Improve Laravel Application Security Further. <http://www.omniceps.com/security-features-laravel-application-security/>. Date of access: 06 Sep 2018.

KRISSHAWEB. 2015. 7 key features of Laravel 5 PHP framework. <https://www.krishaweb.com/7-key-features-of-laravel-5-php-framework/>. Date of access: 06 Sep. 2018.

Lamo, R. 2013. PhpStrorm and why I use it. <https://lamosty.com/2013/02/12/phpstorm-why-i-use-it/> Date of access: 02 Sep. 2018.

Robins, D. and Holmes, J., 2008. Aesthetics and credibility in web site design. *Information Processing & Management*, *44*(1), pp.386-399.

Anon (2018). "WebApp Design." Retrieved 7 October 2018, 2018, from https://[www.suranacollege.edu.in/surana-pg/pdf/mca/Webapp-design.pdf](http://www.suranacollege.edu.in/surana-pg/pdf/mca/Webapp-design.pdf).

Richards, S. (2016). "Content design vs editorial." Retrieved 7 October 2018, 2018, from https://contentdesign.london/content-design/content-design-vs-editorial/.