HACETTEPE UNIVERSITY
COMPUTER SCIENCE & ENGINEERING DEPARTMENT

BBM473 - DATABASE MANAGEMENT SYSTEMS LABORATORY PROJECT

PART 2

SPRING 2020

GROUP NAME: EneryShake

GROUP MEMBERS: Nur Bengisu ÇAM - 21627097
Girayhan YILDIRIM - 21527596

SPORTS CENTER MANAGEMENT SYSTEM

1. PROJECT DEFINITION

Sports Center Management System is an application for sports centers to register Managers, Instructors, Customers and other sports center entities. In the Sports Center Management System, there is an entity set called “SportsCenter”. This entity has all the other entities. For different sports centers there are different entities in the system such as each sports center can define its own package and activities. Entity set “User” defines all the people in the system . Entity set “Role” defines the role information such as “manager”, “instructor” or “customer”. User entity has an attribute as “role” to define whether a user is a “manager”, an “instructor” or a “customer”. Entity set “Service” holds all the sport activities, spas and massages. Entity set “Package” contains the services in that “SportsCenter“ entity. Each SportsCenter entity offers different “packages”. So the entity “packages” are defined by the entity “SportsCenter”. Entity set “Room” holds the physical room information. Entity set “Section” holds the room and service information as well as the time information and creates a “section” entity for the system. Once a user buys a package, he/she can attend any of the opened sections in the package if the section capacity does not exceed the room capacity. Entity set “News” holds the posted news/informations/motivative sentences by an instructor. Only an instructor can post news. Entity set “Address” holds the address information of all the users in the sports center. Entity set “Country” and “City” holds the country and city information, respectively.

2. APPLICATION & ENTITY INFORMATIONS

2.1 SIGN UP

In the Energy Shake - Sports Center Management System, there is a sign up form for the Customers to sign up to the sports center. The Manager is already exists in the database. Instructors are added by Manager. So this sign up form is only for Customers and the role field in the User table is filled as ‘Customer’ once any one signs up. This is because we did not want to give ability to customers to sign up as Instructor or Manager.
We used our custom designed sign up form and verified fields. Once the customer signs up, the page is directed to log in form. Each password is encrypted with SHA-256 and stored like so.

Here are the first entrance page and sign up form page:

Image 1. Entrance Page

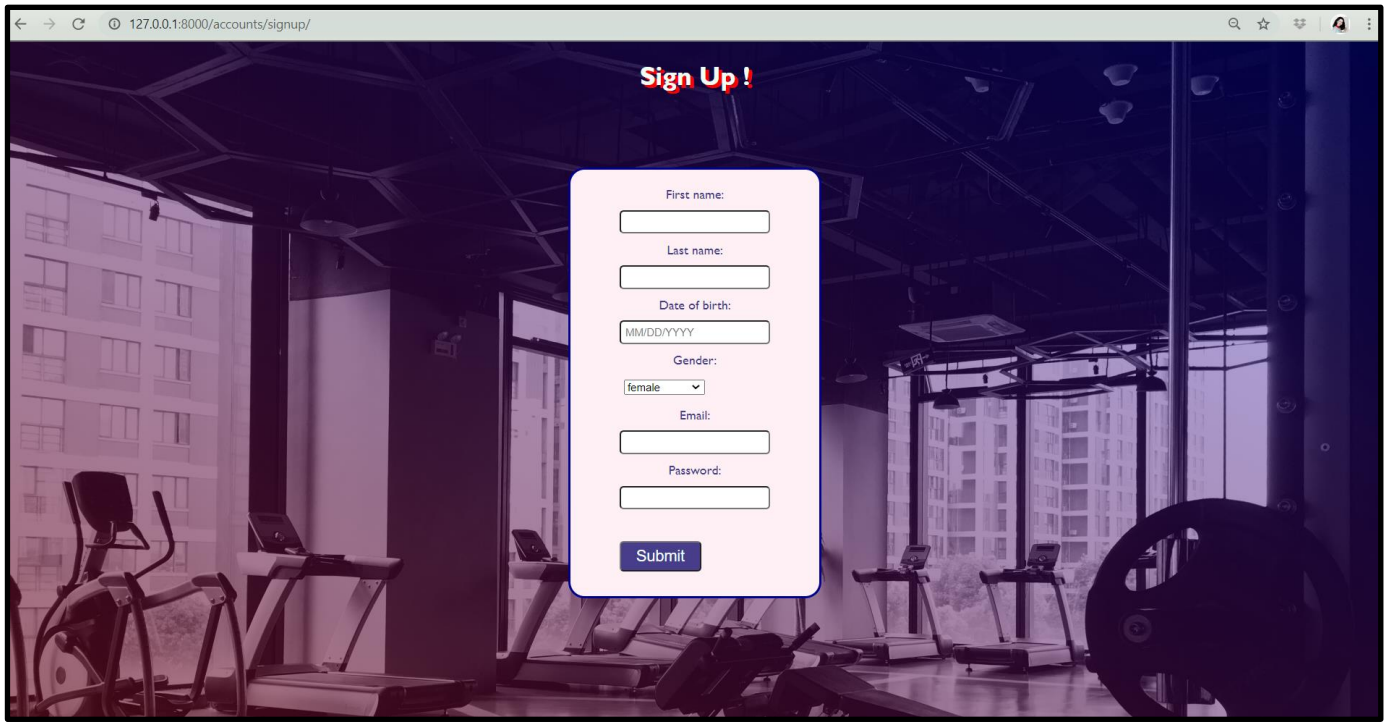


Image 2. Sign Up Page

2.2 LOGIN

Unlike sign up form, every User in the system can log in to the system. User identification and verification is done through ‘email’ and ‘password’ fields.

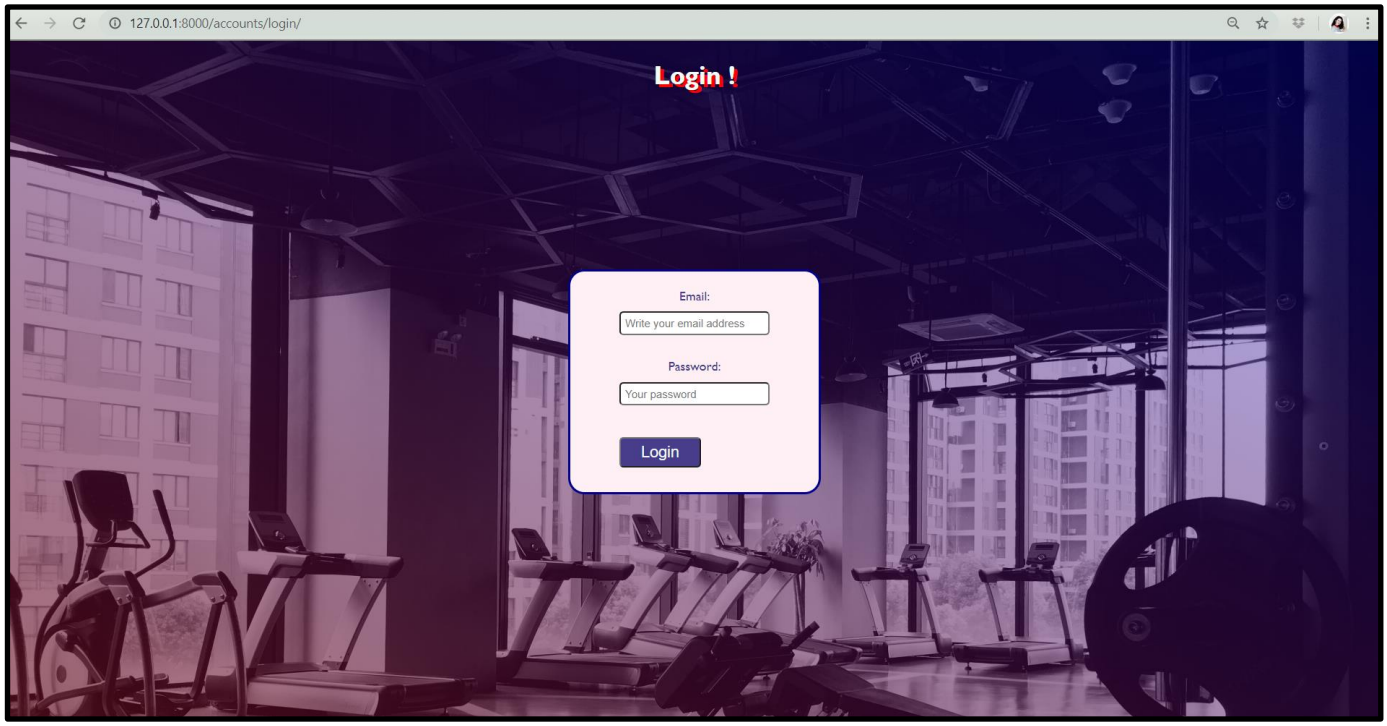


Image 3. Login Page

If the user tries to enter into the system without specifying email and password, system shows a message on the page. Likewise, if the email does not exists or the password is wrong the system shows different messages indicating the mistakes.

2.3 MANAGER

Managers already exist in the system. They can log in with login page. Here is the home page after a successful login:

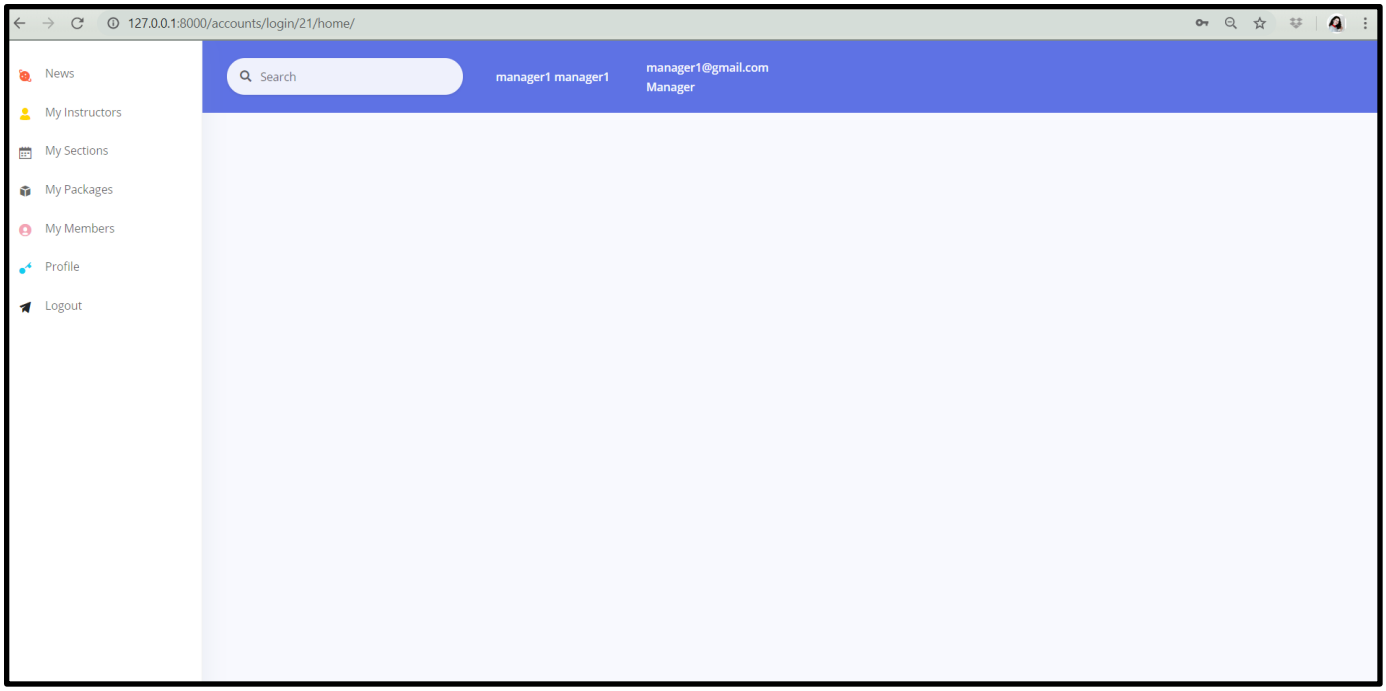


Image 4. Home

On the left side, there is a navigation bar. As you can see, a manager can access News, Instructors, Sections, Packages, Members and his/her own profile information. There is a logout button on the left which lets you log out of the system and redirects you to login page. In the url, we hold the logged in user's 'user_id' and from that information we can get the active user information when it is needed.

2.3.1 News

A manager can views the News written by instructors.

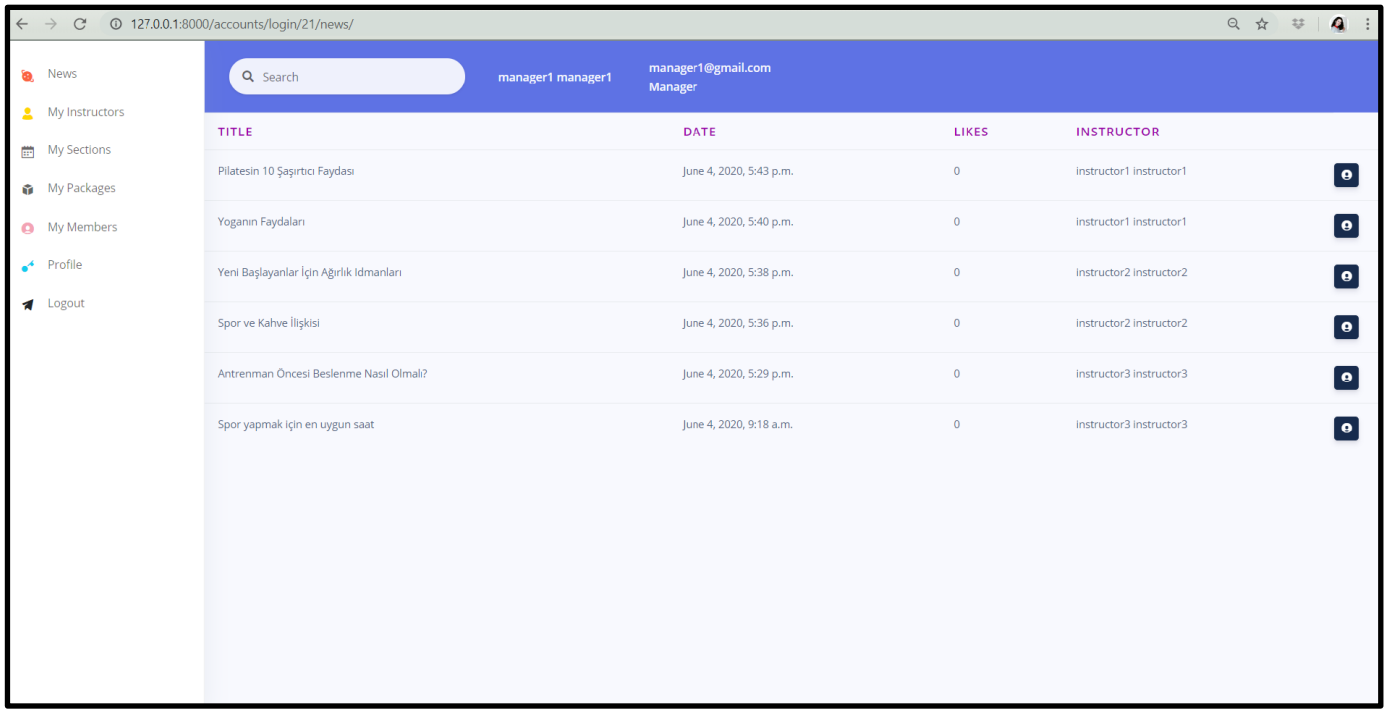


Image 5. Listing News

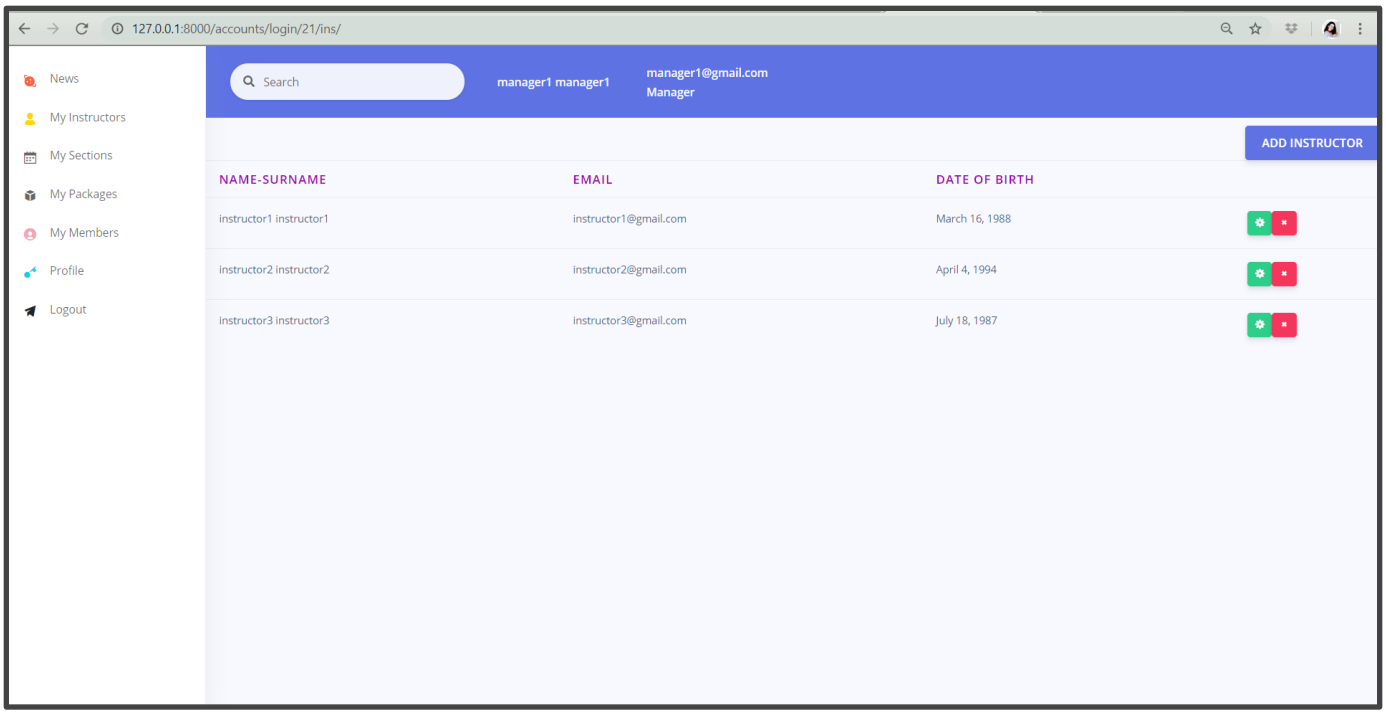


Image 6. Listing Instructors, Update and Delete Buttons

2.3.2 My Rooms

A manager can list, update, delete and add rooms for their sports center.

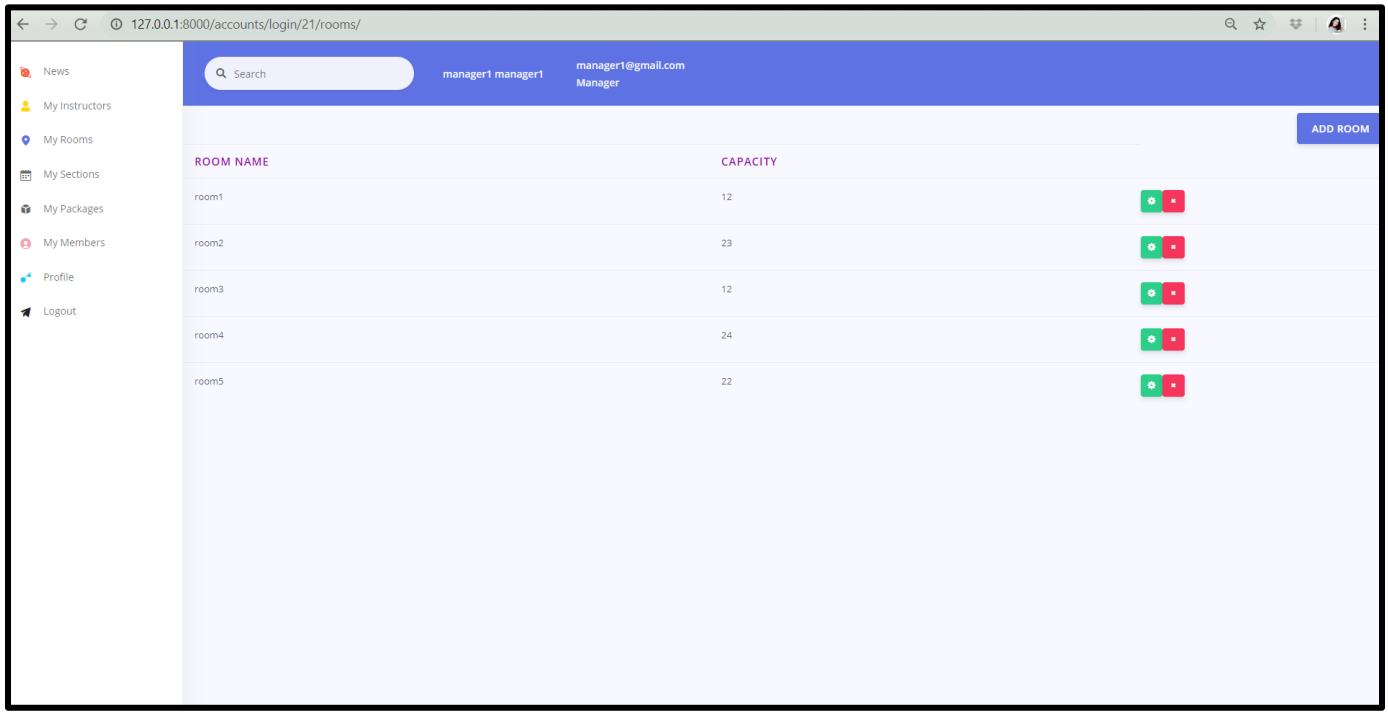


Image 7. Listing Rooms

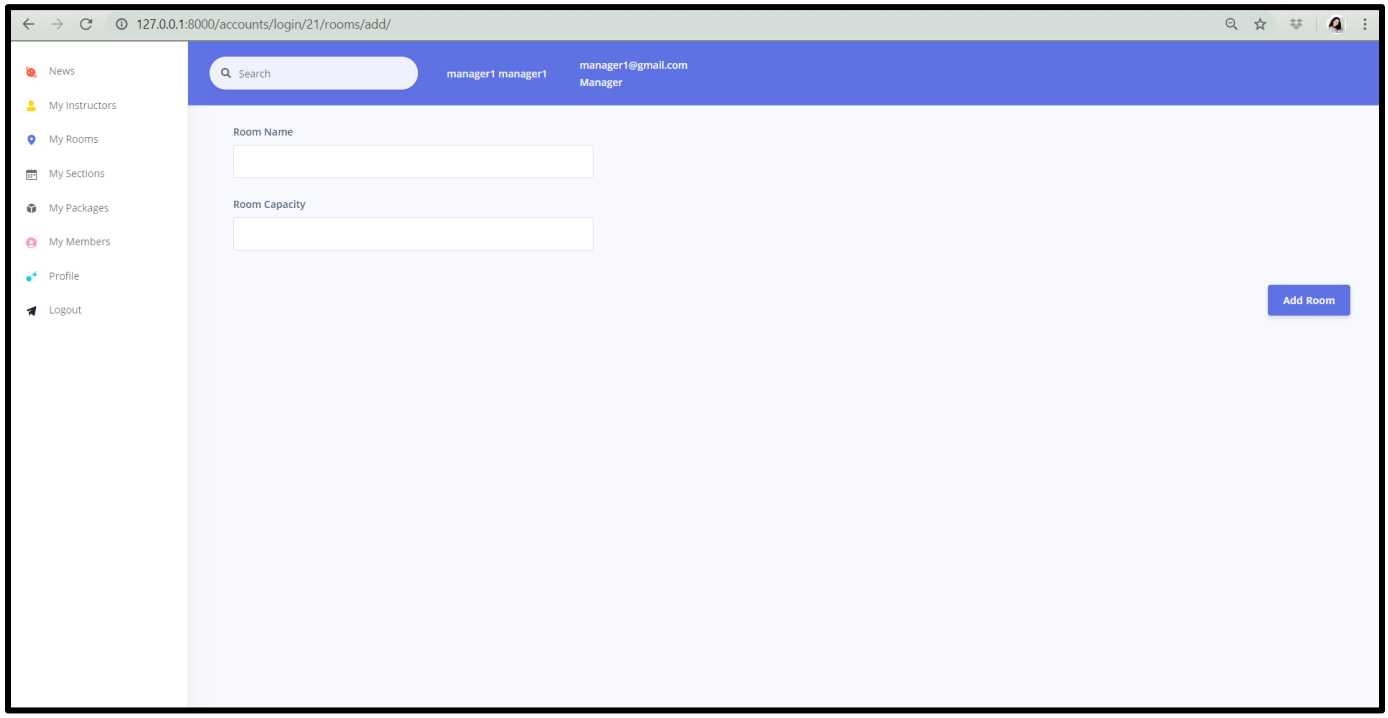


Image 8. Adding Room

2.3.3 My Instructors

A manager can list, update and delete the instructors in the system. From the green button on right, manager can update and from the red button on right

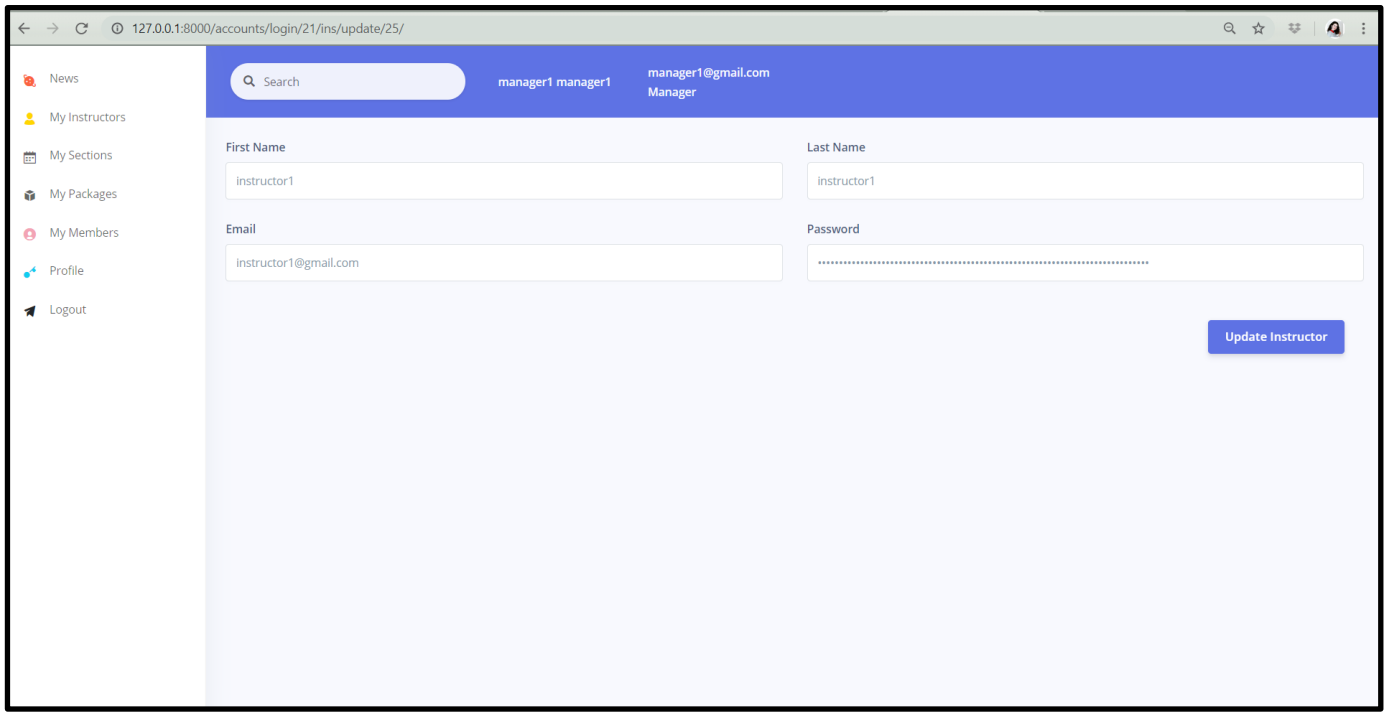


Image 9. Updating Instructor

2.3.4 My Sections

A manager can list, create, update and delete sections in the system.

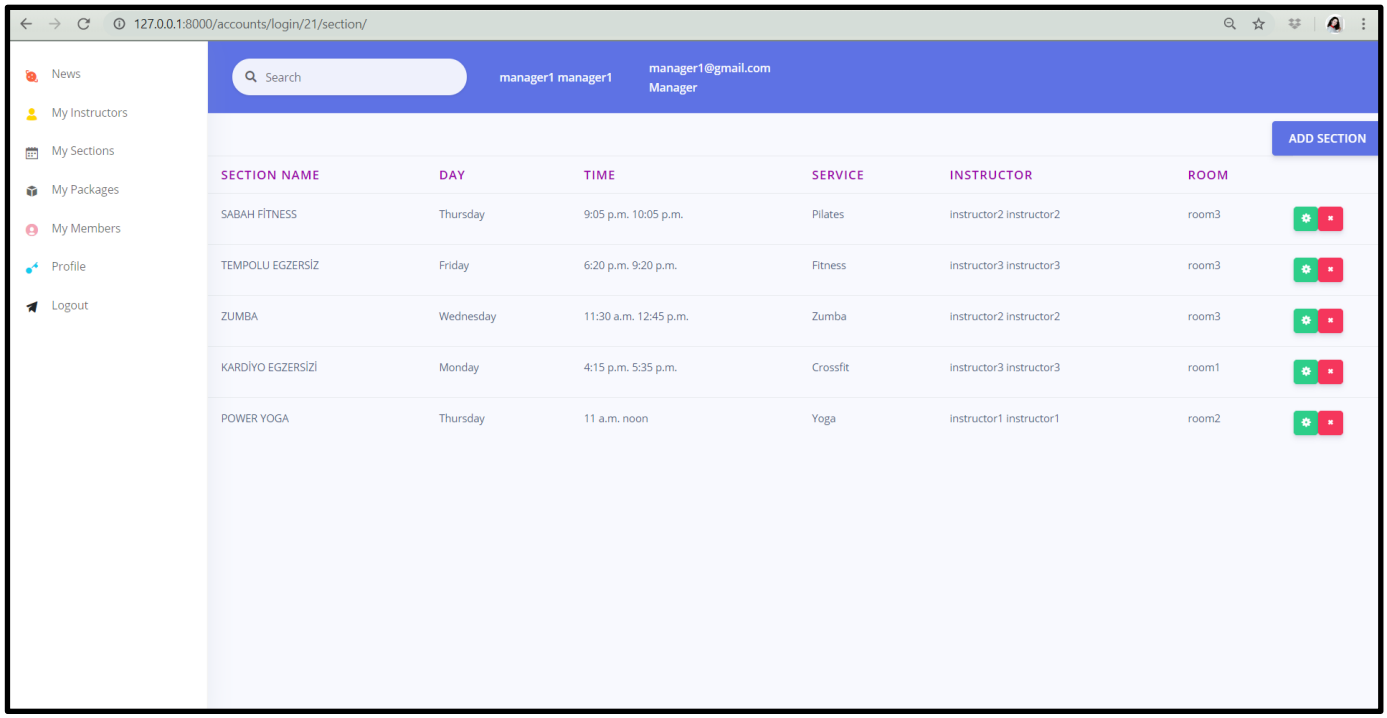


Image 10. Listing Sections

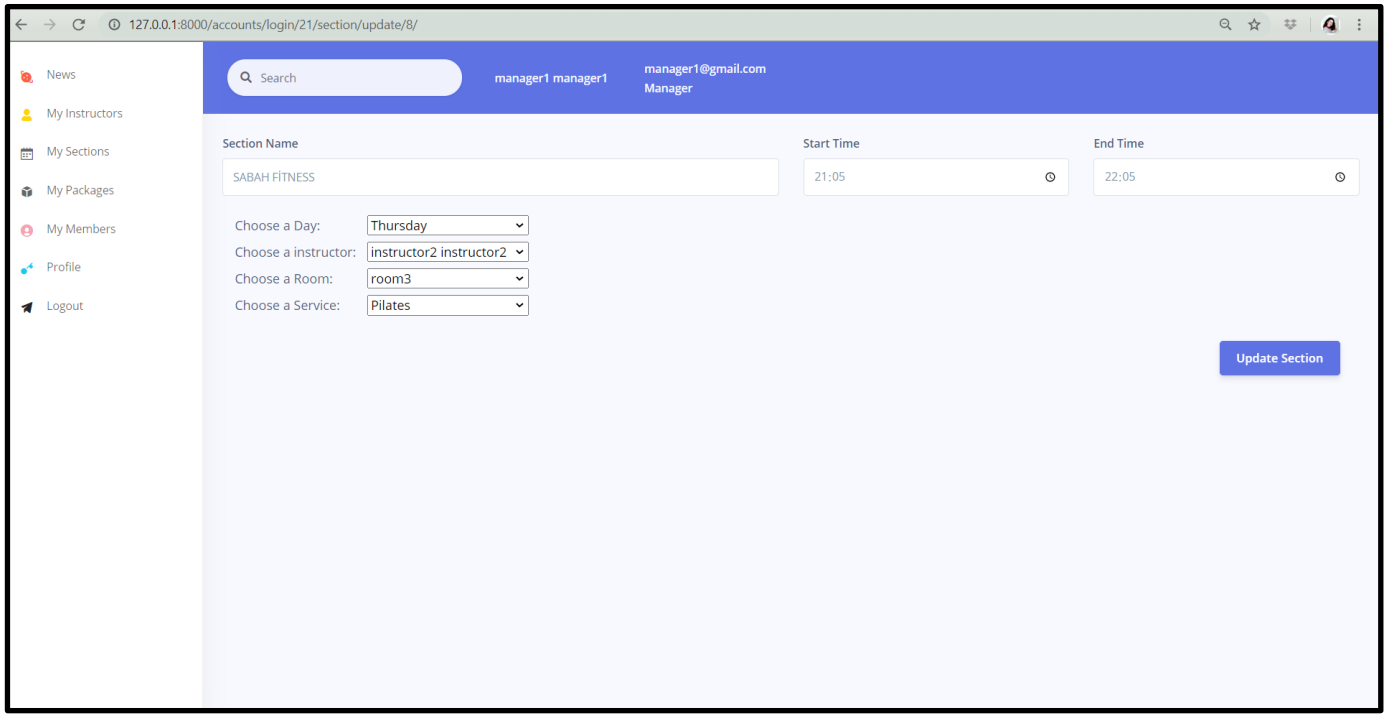


Image 11. Updating Section

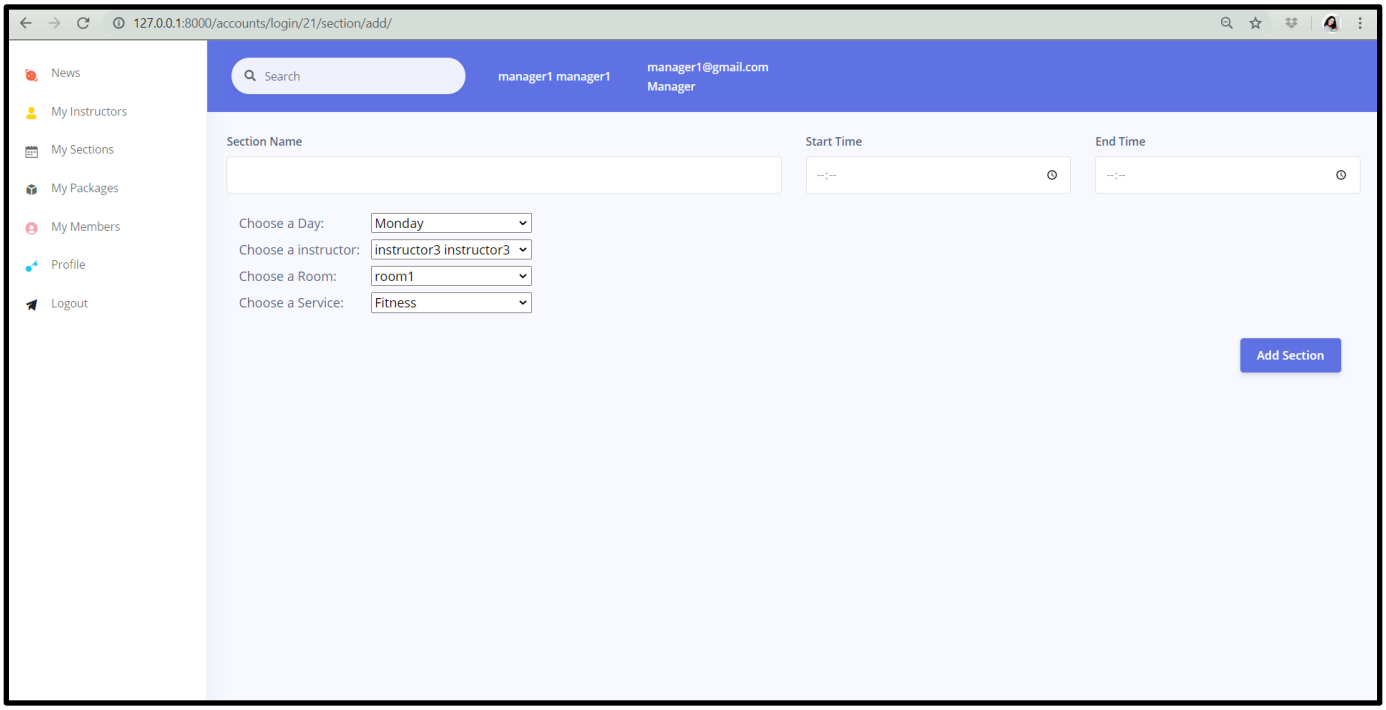


Image 12. Creating Section

2.3.5 My Packages

A manager can list, create, update and delete packages. No one can create, update and delete packages other than the manager.

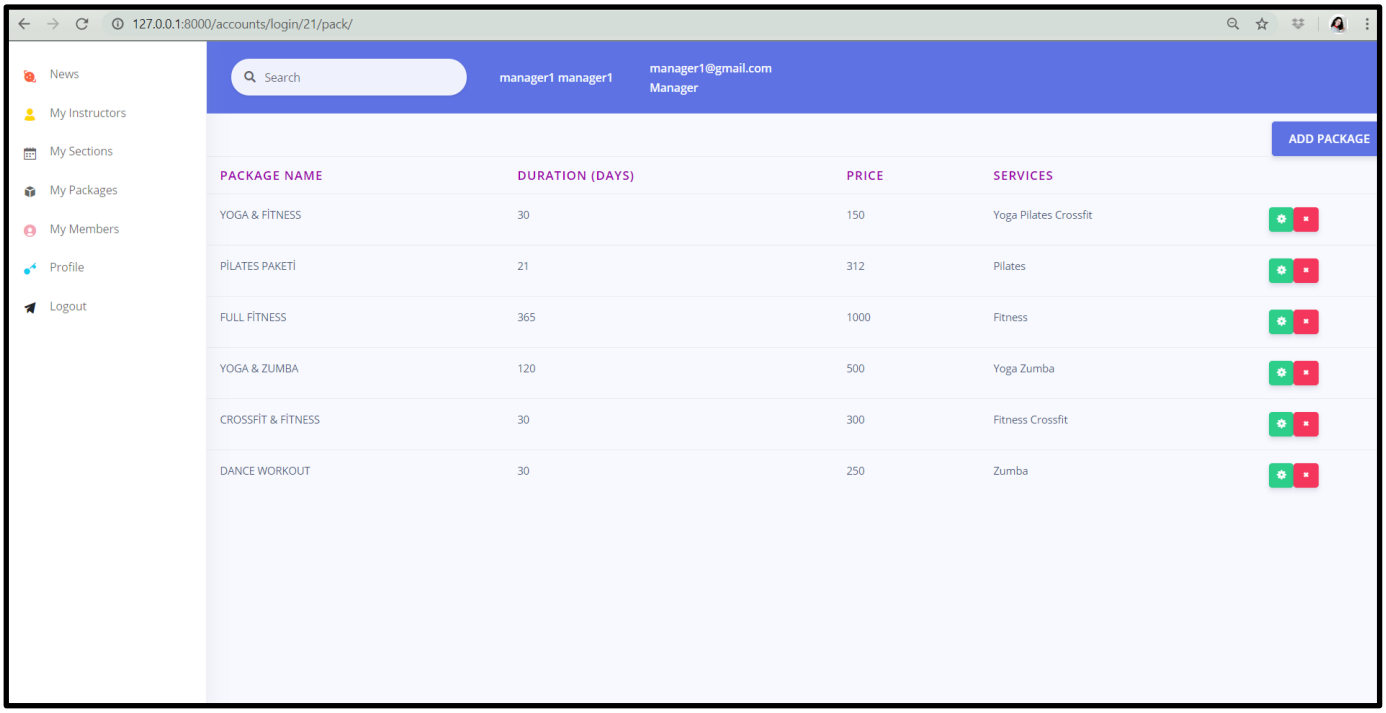


Image 13. Listing Packages

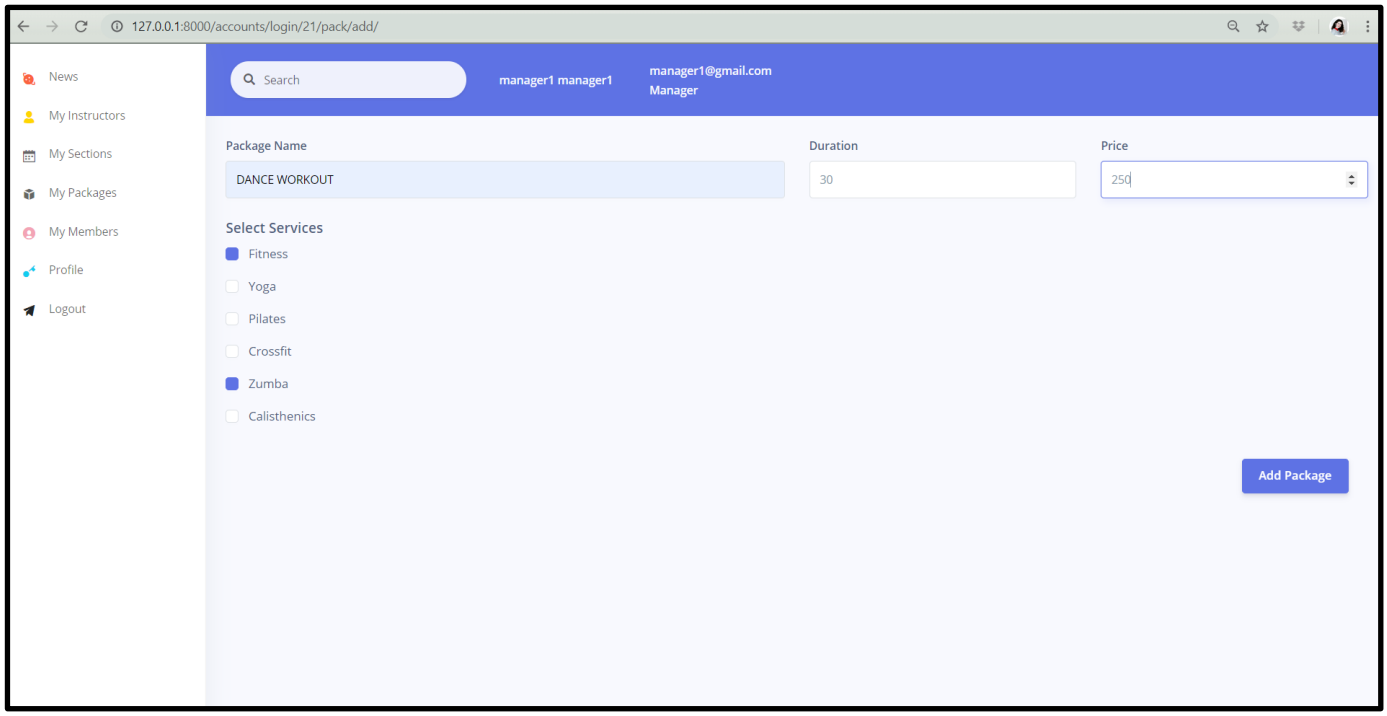


Image 14. Adding Packages

2.3.6 My Members

A manager can list all the customers who are enrolled(bought a package) into his/her sports center.

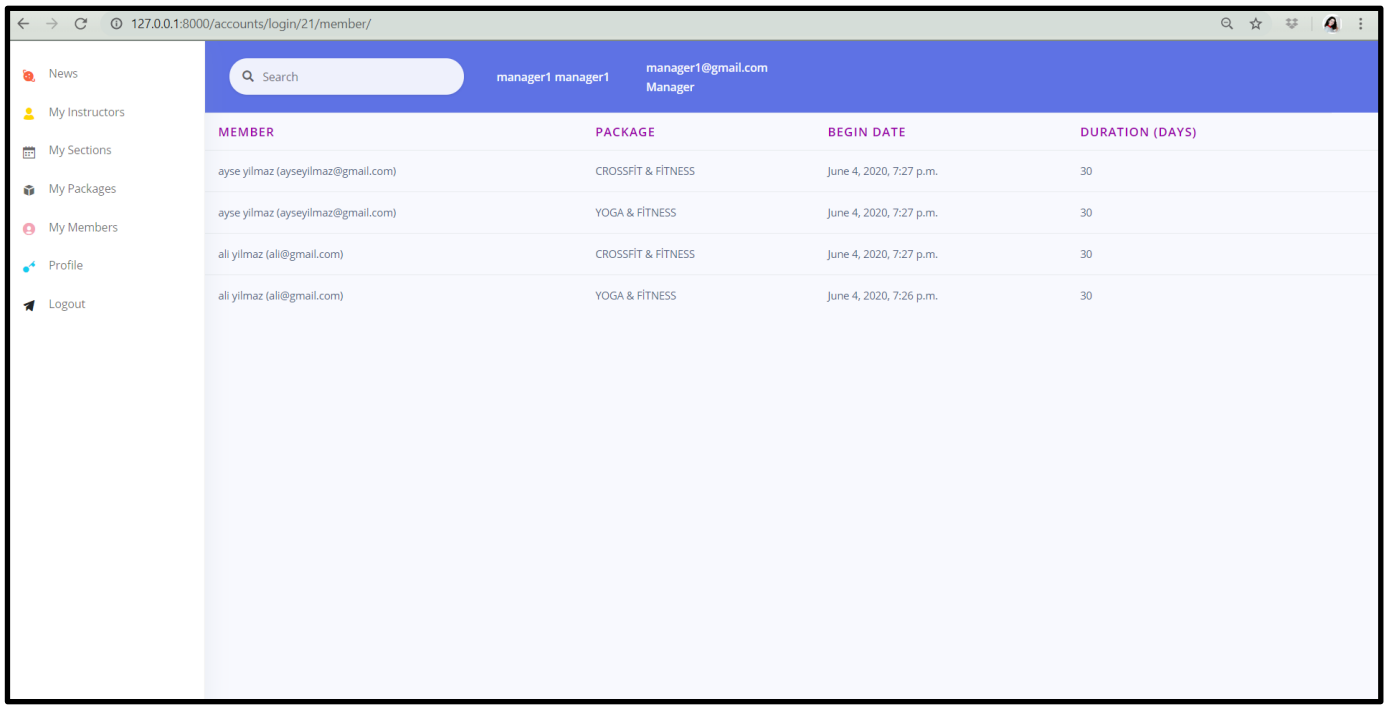


Image 15. Listing Members

2.4 INSTRUCTOR

2.3.1 News

An instructor is created by a manager. Instructor then can log in to the system. Only instructor can create, update news.

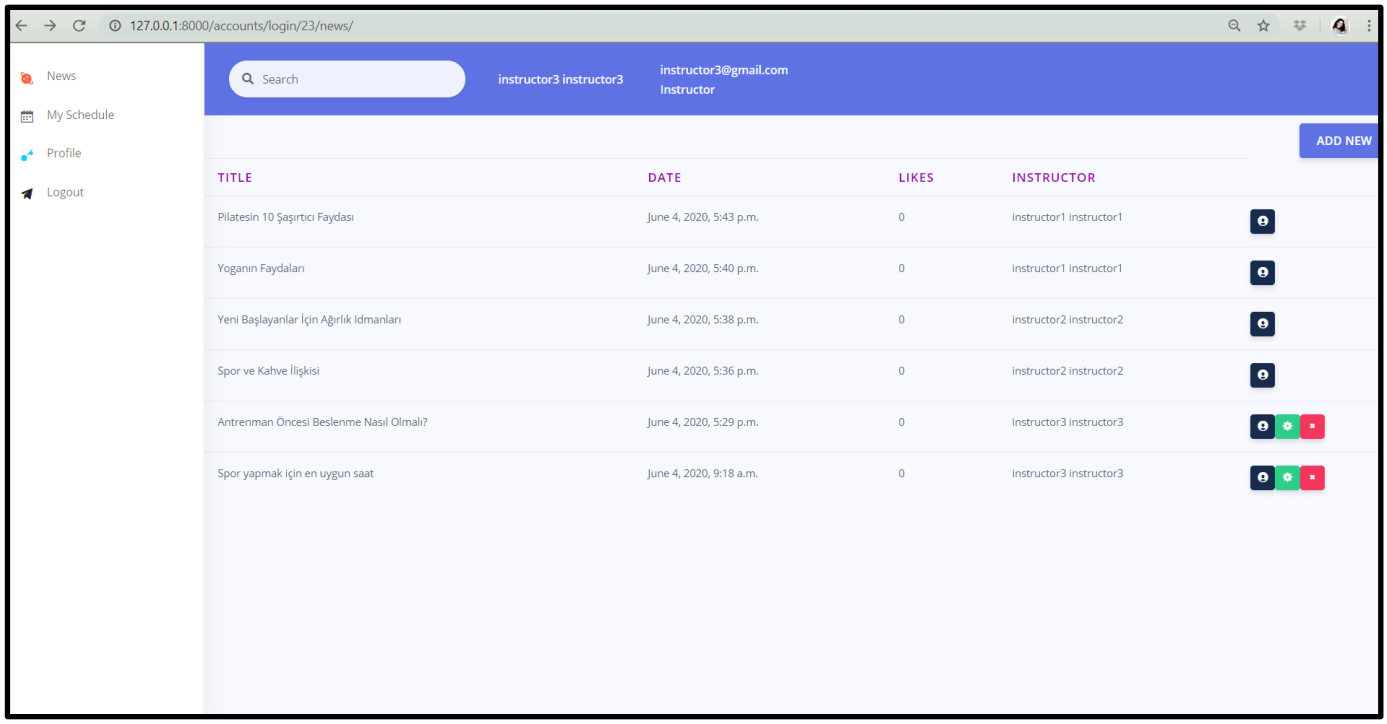


Image 16. Listing News

Image 17. View News Detail

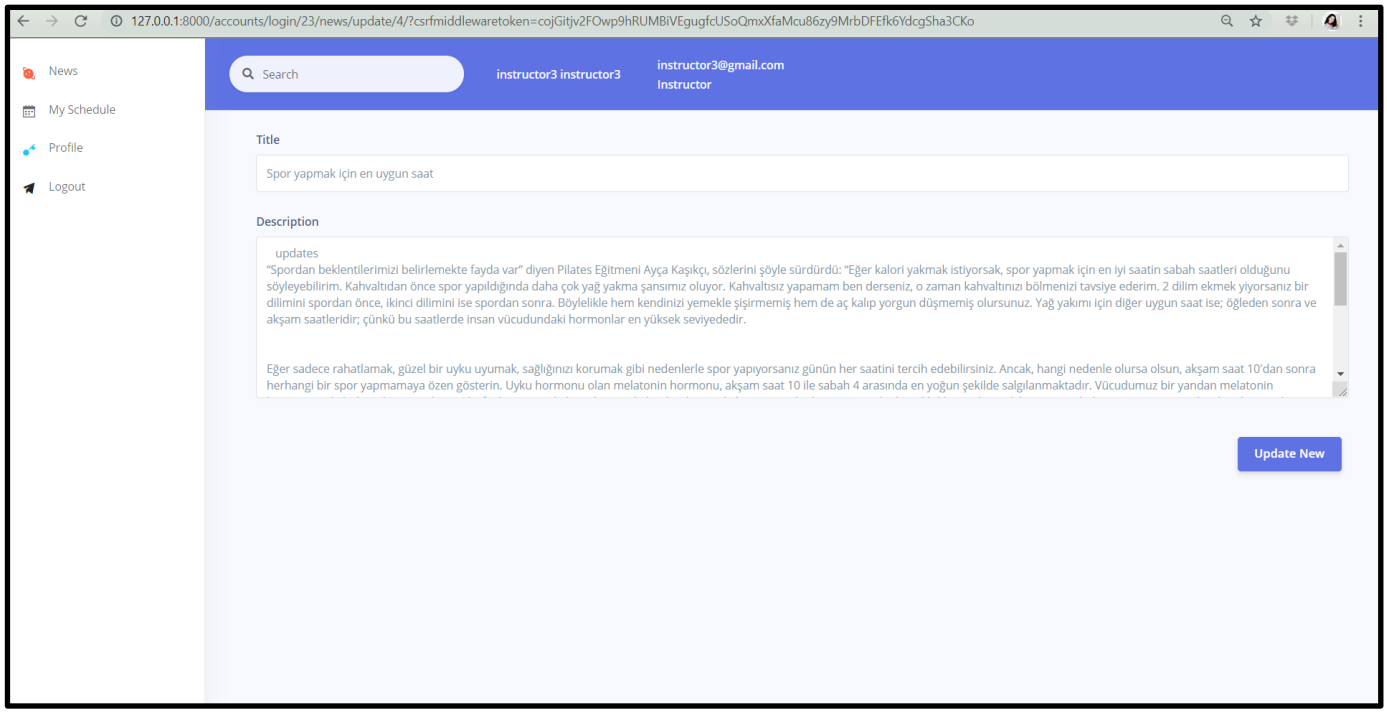


Image 18. Updating News

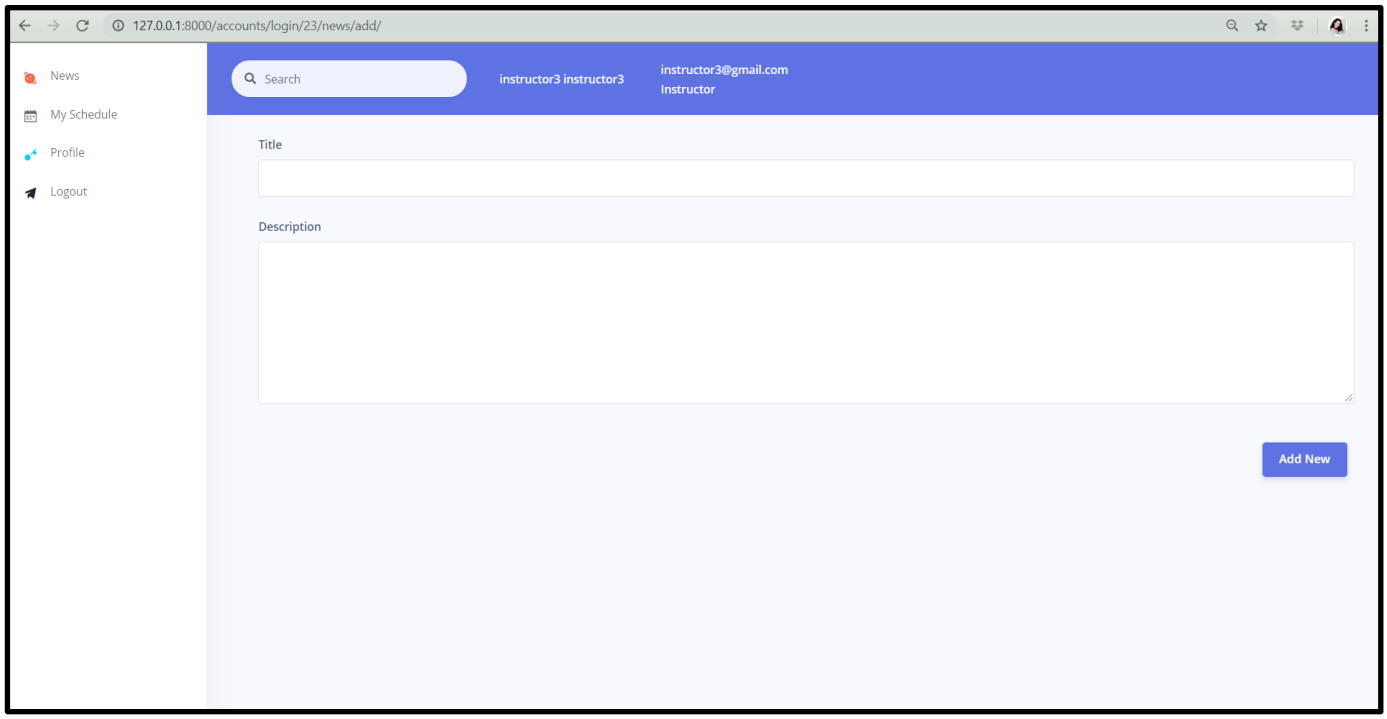


Image 19. Create News

2.3.1 My Schedule

An Instructor can view his/her schedule.

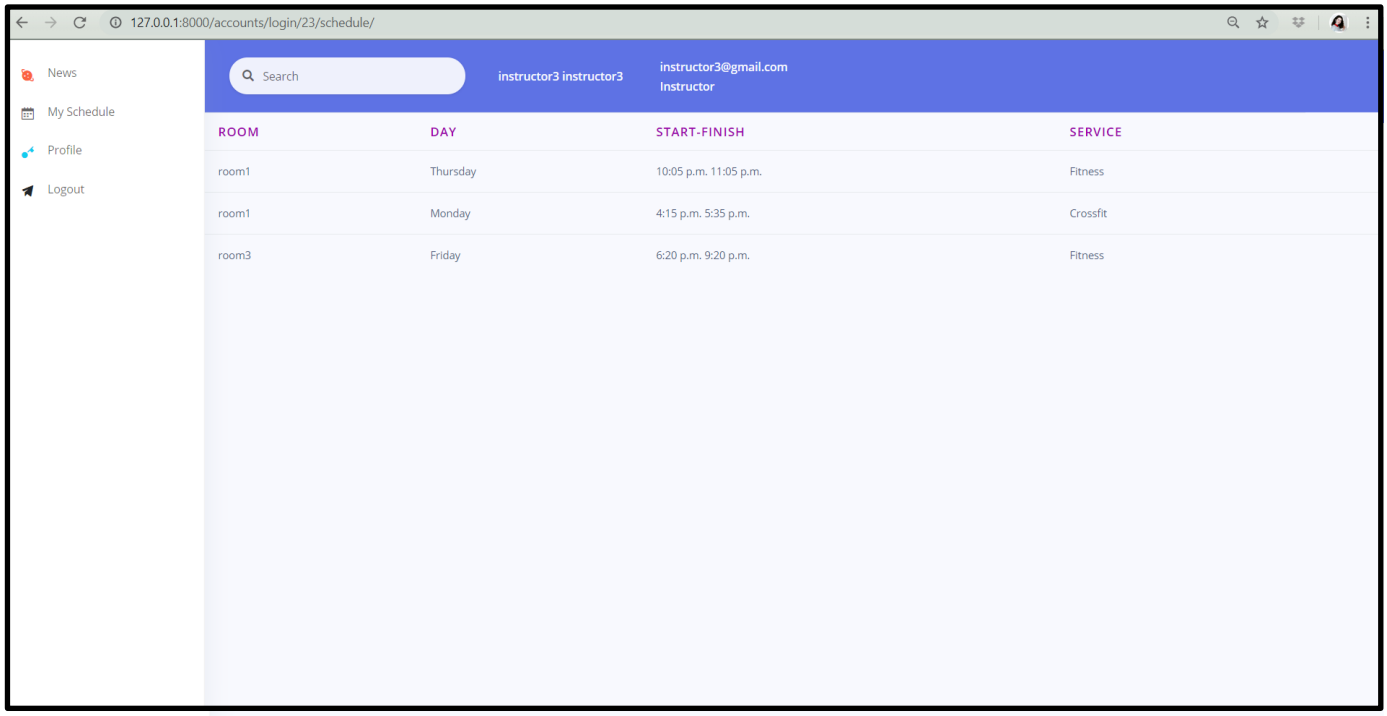


Image 20. View Schedule

Every User in the system, can view his/her profile. From the button on the right, user can change password by verifying the new password.

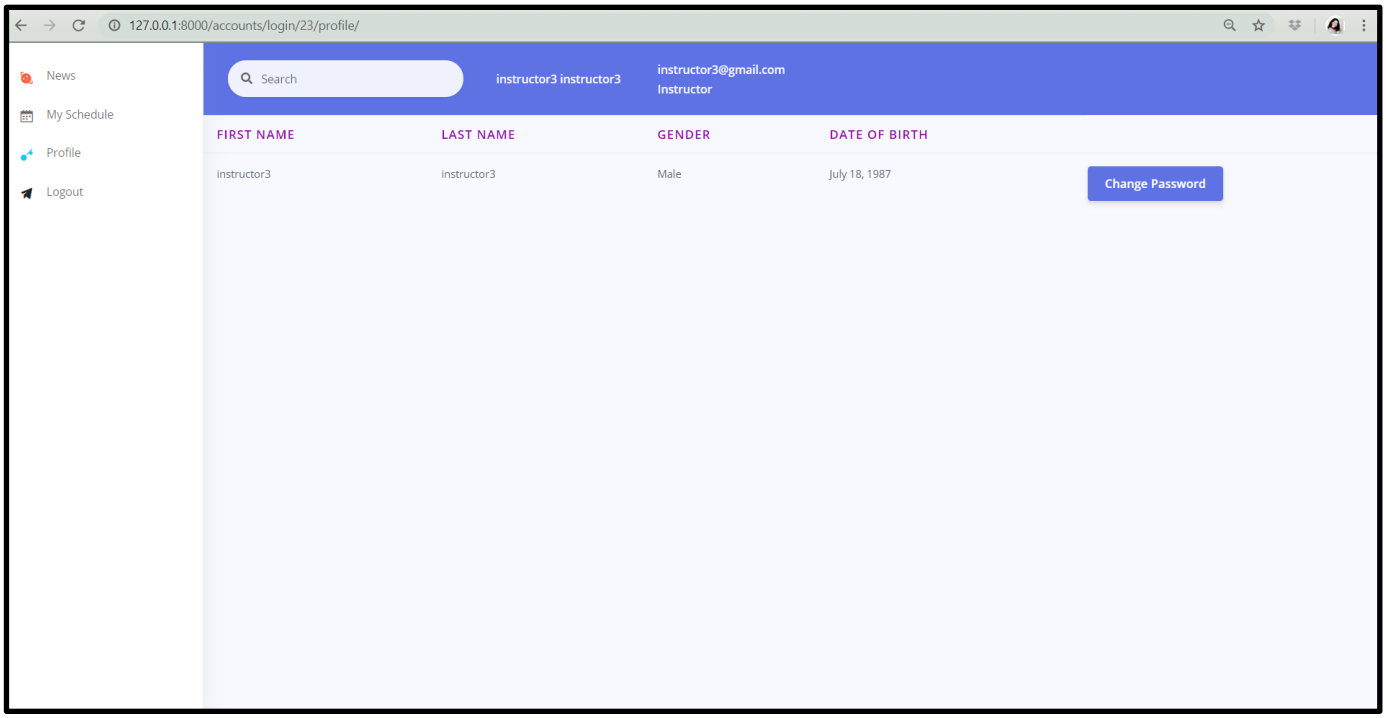


Image 21. Profile

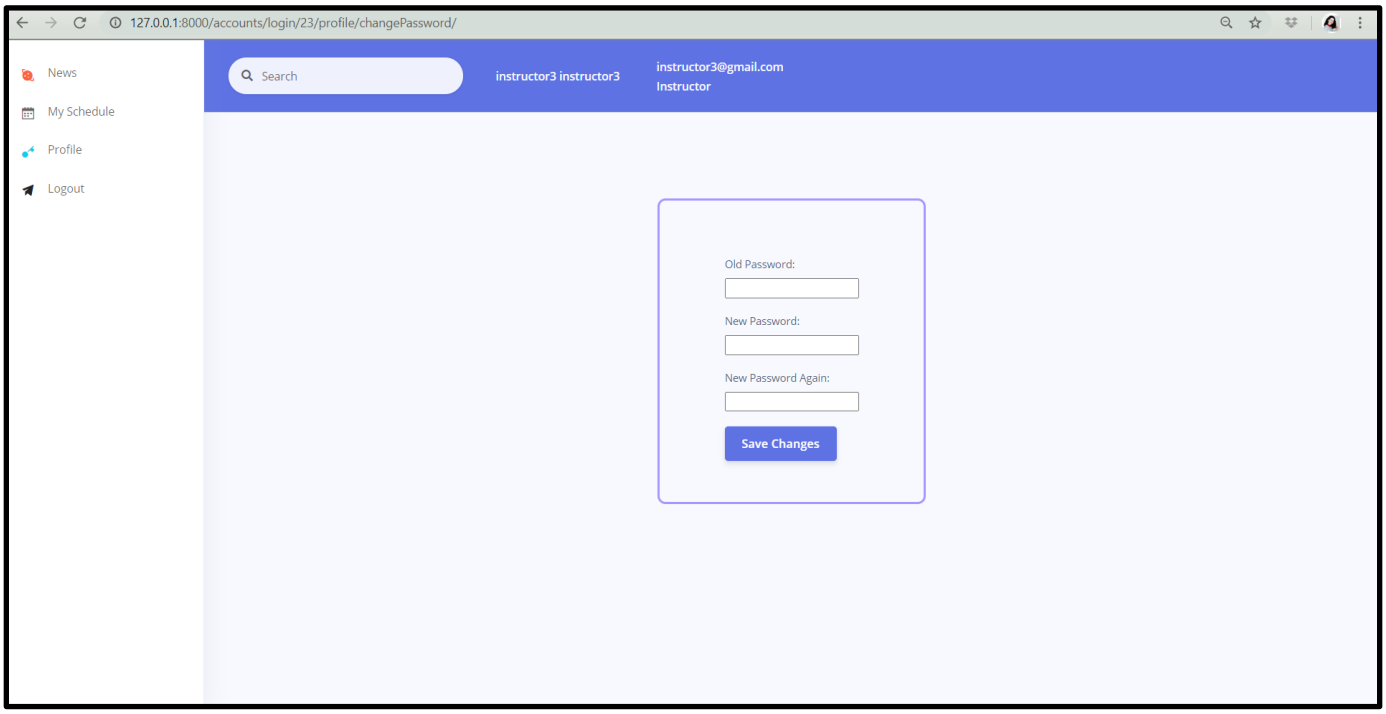


Image 22. Change Password

2.5 CUSTOMER

2.5.1 News

A customer can only list and view the details of the news which are created by the instructors.

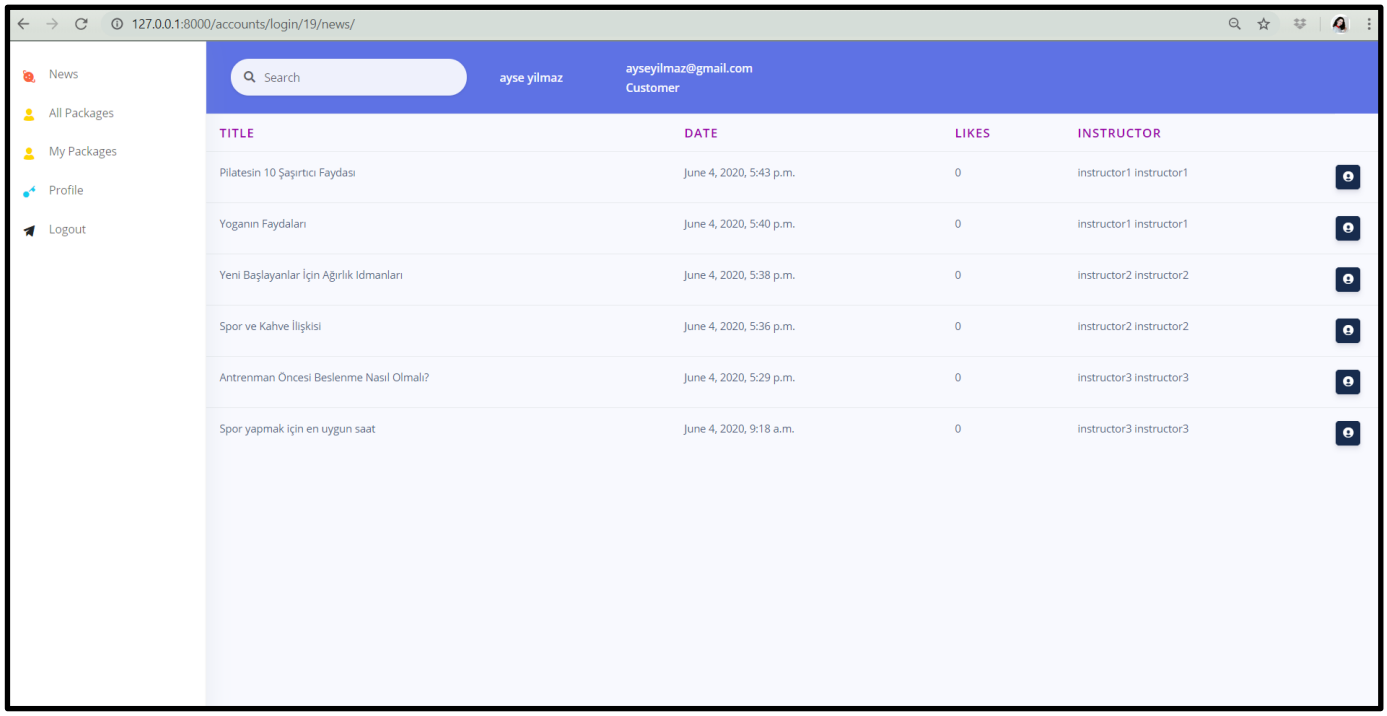


Image 21. Listing the News

Image 22. Details of the News

2.5.2 All Packages

A customer can list all the packages that exist in the system. With the button on the right, he/she can buy a package or more.

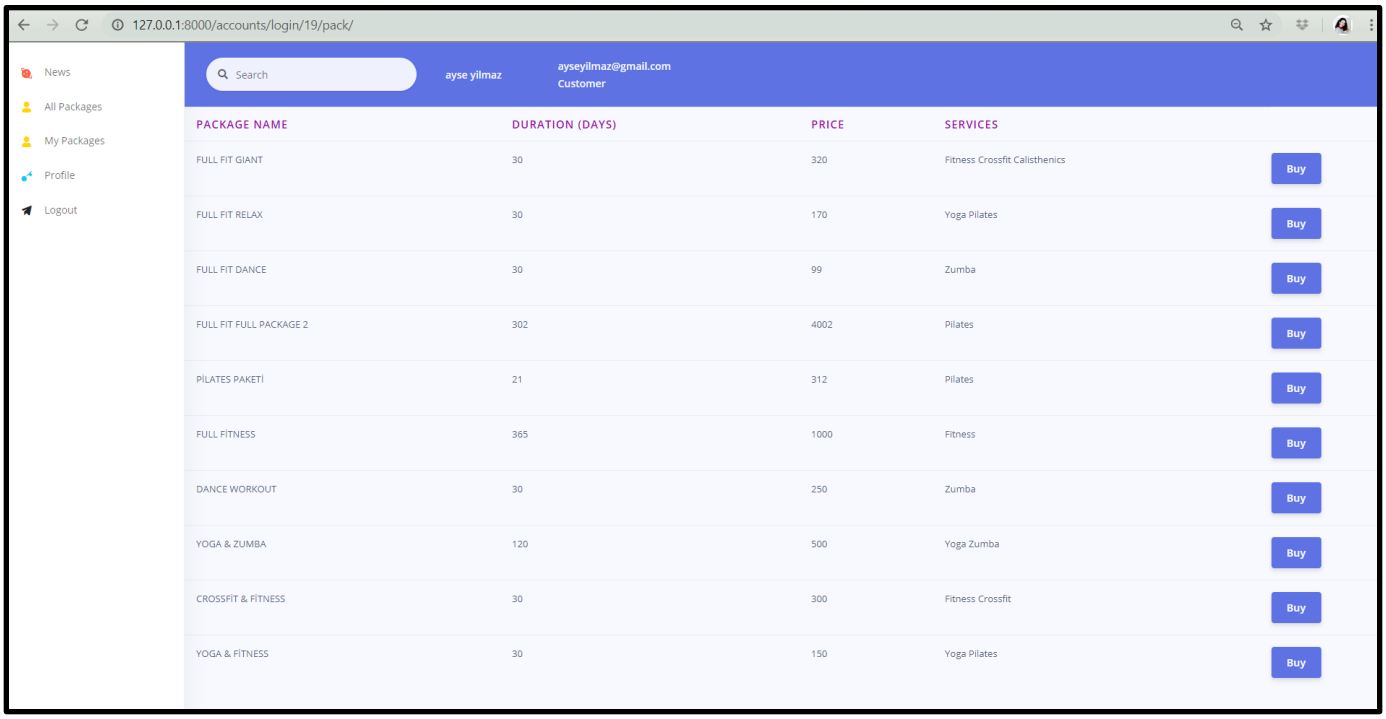


Image 23. Listing Packages

2.5.2 My Packages

A customer can list the packages that he/she bought.

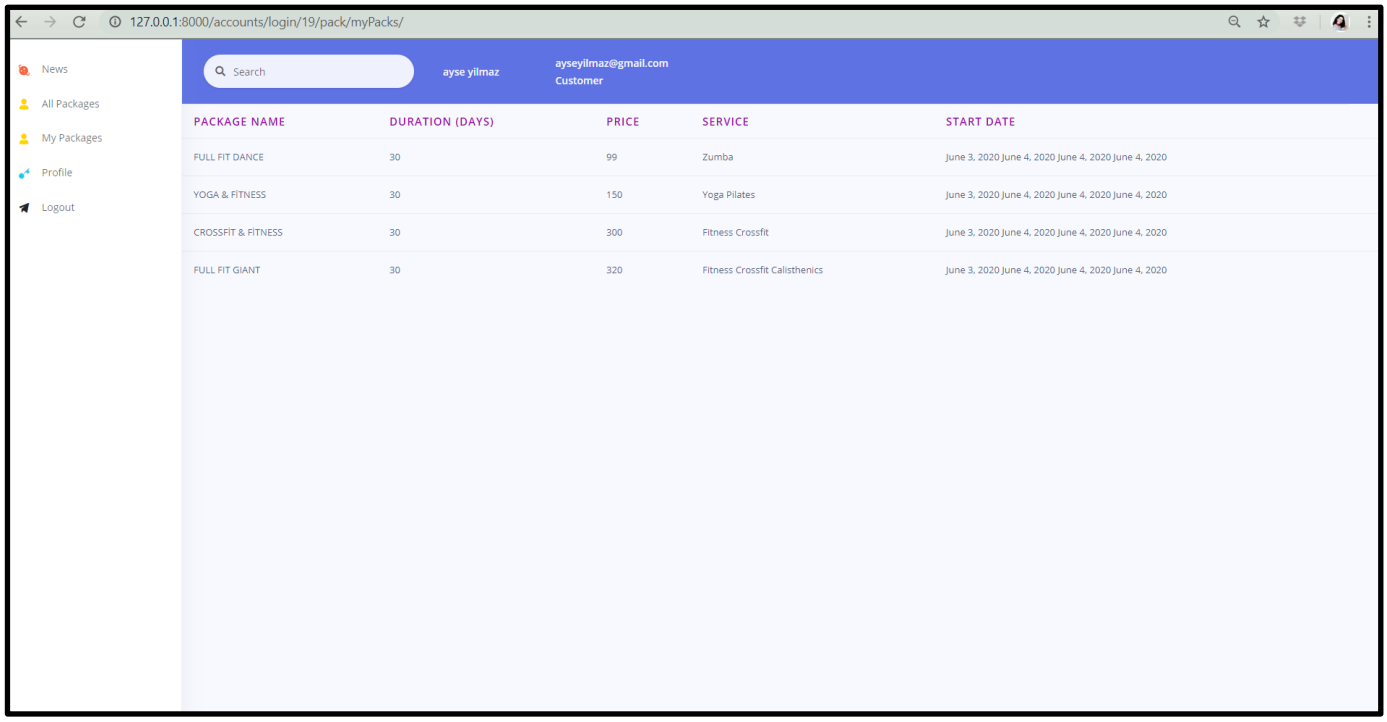


Image 24. My Packages

3. REQUIREMENTS & CONSTRAINTS

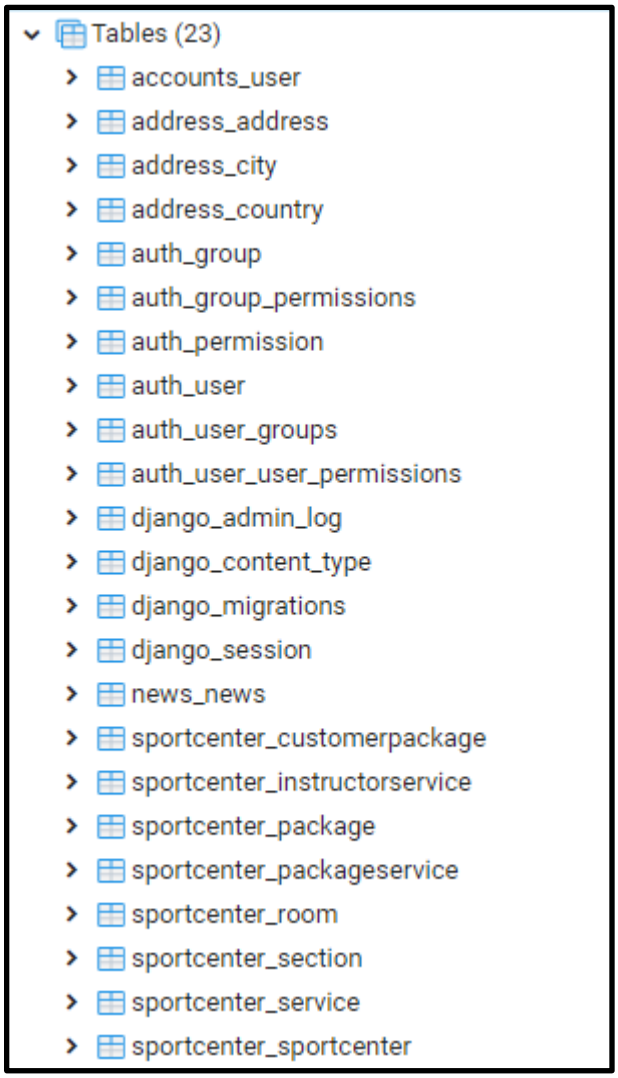
- Sport Center must have at least one instructor and one manager.
- There should be only one instructor in the Section.
- Only one service can be provided in the Section.
- Sport Center must have at least one package.
- Package must contain at least one Service.
- Sport Center must have an Address.
- News can only be written by an Instructor.

In the application, there are a lot of input fields and forms. They are required to be filled by the user. To be able to prevent submitting empty data, we checked the fields. If the necessary fields are not filled, then the system renders a message to the user. In almost every html file and views, we used these checks to guide the user. Here is an example code:

```
def change_password_action(request, user_id):
    active_user = User.objects.get(pk=user_id)
    password = request.POST.get('old_password')
    if len(password) > 0 and check_password(password, active_user.password):
        new_password = request.POST.get('new_password')
        new_password_again = request.POST.get('new_password_again')
        if len(new_password) > 0 and len(new_password_again) > 0:
            if new_password == new_password_again:
                active_user.password = make_password(new_password)
                active_user.save()
            else:
                messages.error(request, 'New passwords do not match!')
                return render(request, 'sportcenter/change_password.html',
                              {'active_user': active_user})
        else:
            messages.error(request, 'Please fill the blanks!')
            return render(request, 'sportcenter/change_password.html',
                          {'active_user': active_user})
    else:
        messages.error(request, 'Password is not correct!')
        return render(request, 'sportcenter/change_password.html',
                      {'active_user': active_user})

    return render(request, 'sportcenter/profile.html',
                  {'active_user': active_user})
```

4. TABLES

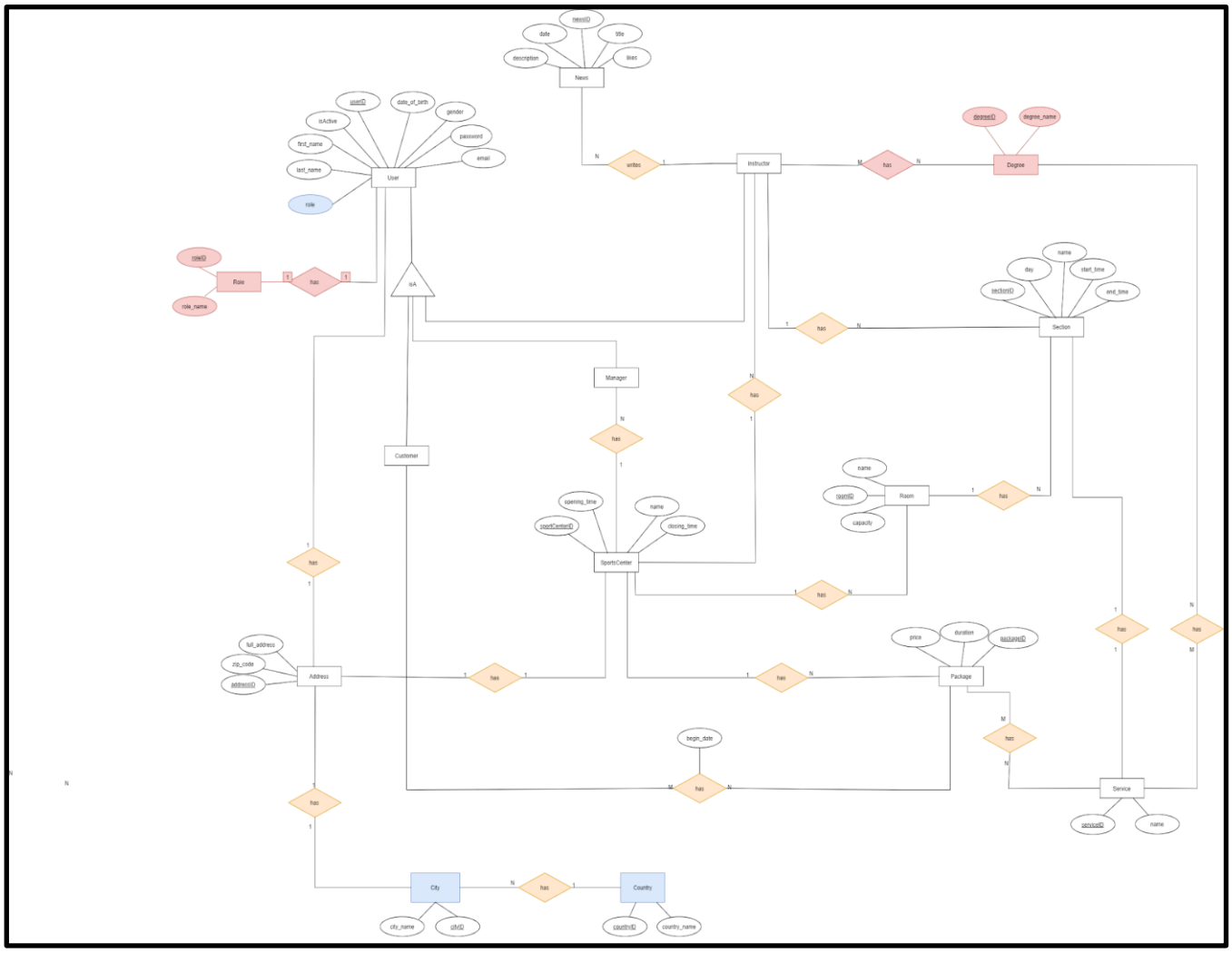
As previously mentioned, we used postgresql as a database.Our Relation Tables are like this. Also, these tables and their features are shown in the diagram below.



```
CREATE TABLE public.sportcenter_sportcenter
(
    id integer NOT NULL DEFAULT nextval('sportcenter_sportcenter_id_seq'::regclass),
    sport_center_name character varying(100) COLLATE pg_catalog."default" NOT NULL,
    opening_time time without time zone NOT NULL,
    closing_time time without time zone NOT NULL,
    address_id_id integer NOT NULL,
    CONSTRAINT sportcenter_sportcenter_pkey PRIMARY KEY (id),
    CONSTRAINT sportcenter_sportcen_address_id_id_a08ca636_fk_address_a FOREIGN KEY (address_id_id)
        REFERENCES public.address_address (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        DEFERRABLE INITIALLY DEFERRED
)

TABLESPACE pg_default;

ALTER TABLE public.sportcenter_sportcenter
    OWNER to postgres;
-- Index: sportcenter_sportcenter_address_id_id_a08ca636

-- DROP INDEX public.sportcenter_sportcenter_address_id_id_a08ca636;

CREATE INDEX sportcenter_sportcenter_address_id_id_a08ca636
    ON public.sportcenter_sportcenter USING btree
    (address_id_id ASC NULLS LAST)
    TABLESPACE pg_default;
```

Table 1. Example

5. RELATIONS



The red color means drop the relations, entity or attributes.
The blue color means change the relations, entity or attributes.

Diagram links for high resolution:
https://drive.google.com/file/d/15JX4KzggBYfe8CVjpUZ2xUhQOyKJ3LZt/view?ts=5e823345

One-to-one relationships:

-User Address: User can only have one address.
-Sportscenter-Address: Sportscenter can only have one address.
-Address-City: The address can contain only one city.
-Section-Service: Section can give only one service

One to many relations:

-Instructor-News :We choose one to many relation for instructor-news relation because an instructor can write many news into the system.

-Country-City: We choose one to many relations for country-city relations because countries have many cities.

-Instructor-Section: We choose one to many relation for instructor-section relation because instructors can join the many sections.

-Sportscenter-Manager: We choose one to many relation for spotcenter-manager relation because a sportscenter can has a lot of managers.

-Sportscenter-Instructor: We choose one to many relation for spotcenter-instructor relation because a sportscenter can have a lot of instructors.

-Sportscenter-Package: We choose one to many relation for spotcenter-package relation because a sportscenter can sell a lot of packages.

-Sportscenter-Room: We choose one to many relation for sportscenter-room relation because a sportscenter can has a lot of rooms.

-Room-Section: We choose one to many relation for room-section relation because room can be used by many sections.

Many to many relations:

-Package-Service : We choose many to many relation for package-service relation because a package can include a lot of services and a service can select by a lot of packages.

-Customer-Package: We choose many to many relation for customer-package relation because a customer can buy a lot of packages and a package can be bought by a lot of customers.

-Instructor-Service : We choose many to many relation for instructor-service relation because an instructor can give a lot of services and a service is given by a lot of instructors.

6. RUN THE APPLICATION

You can either sign up as a customer or use one of the users that are already exist in the database. You can login with manager and create instructor and then login with it if you do not want to use the instructor given below. However, you need to use the manager given below since we assume that manager already exists in the database. If we could let users to specify which role are they, that would cause security problems.

Manager email : manager1@gmail.com
Manager password: 123456
Instructor email: instructor3@gmail.com
Instructor password: ins312

/. REFERENCES

- https://docs.djangoproject.com/en/3.0/
- https://www.youtube.com/playlist?list=PL4cUxeGkcC9ib4HsrXEYpQnTOTZE1x0uc
- https://getbootstrap.com/docs/4.1
- https://www.elephantsql.com/