



# Benha Clinics

***Undertaken by:***

Omar Ali Amin El-sherif(99)

Amr Atif Mahmoud(194)

Abdelwahab Mohamed Abdo (96)

Ahmed Tarek El_Sayead (10)

Horria Abdelaleam salh (64)

***Supervised by:***

Dr. Mohammed Abdelfattah

Academic year: 2021-2022

# Contents



## Table of Figures

## Table of Tables

Introduction:

The problem is the Clinics were very crowded and patients wait a very long time to make just one appointment, so we came up with the solution. the patient now can book his appointment online and come to the clinic in his appointment with no queue, on the other hand, the clinic is not crowded and able to be registered on the site and show all appointments. First, we made a site to serve one clinic and its patients, then we generalized the idea. so, we enhanced the site, the site can serve more than one clinic and many patients. Our system is considered a middleware between clinics and patients, and provides the spirit of competition between clinics, and that makes doctor's clinics improve the techniques that they provide to their patients.

UML diagrams:

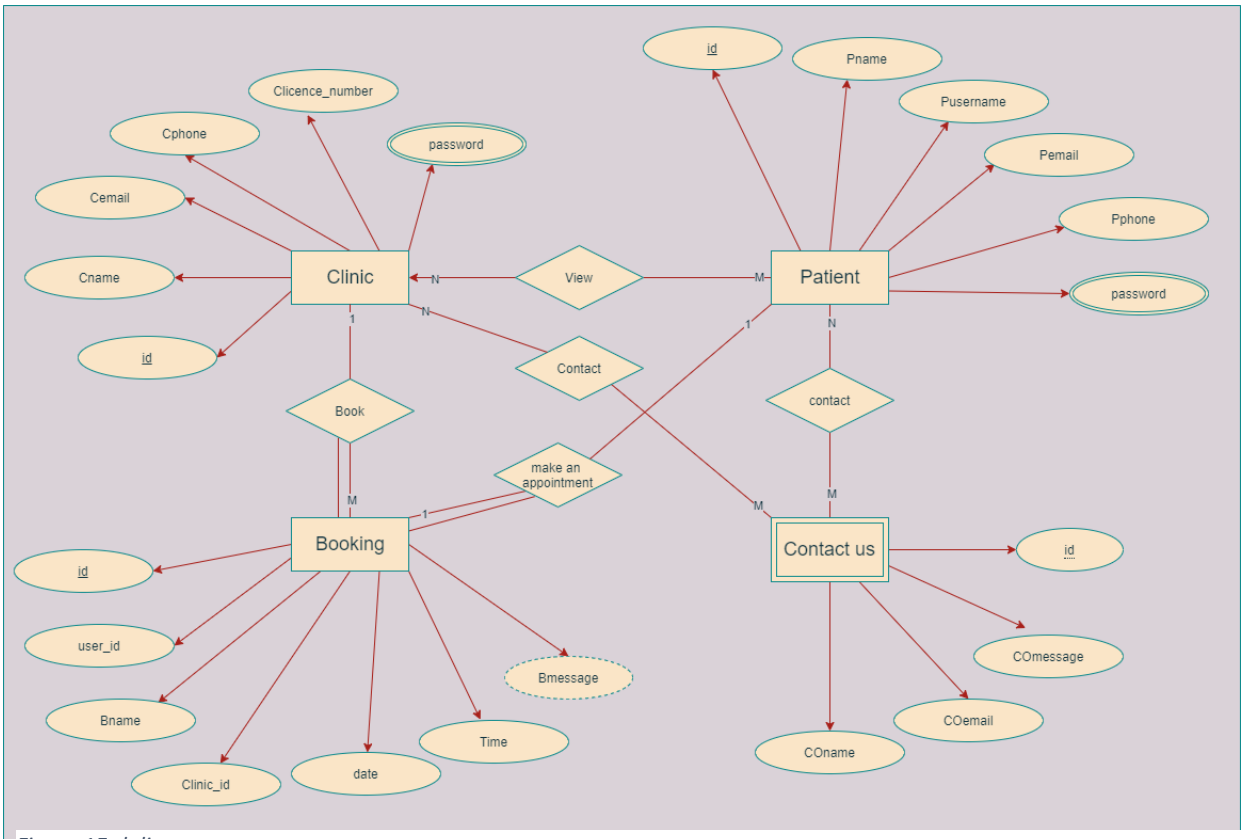


*Figure 1Erd diagram*

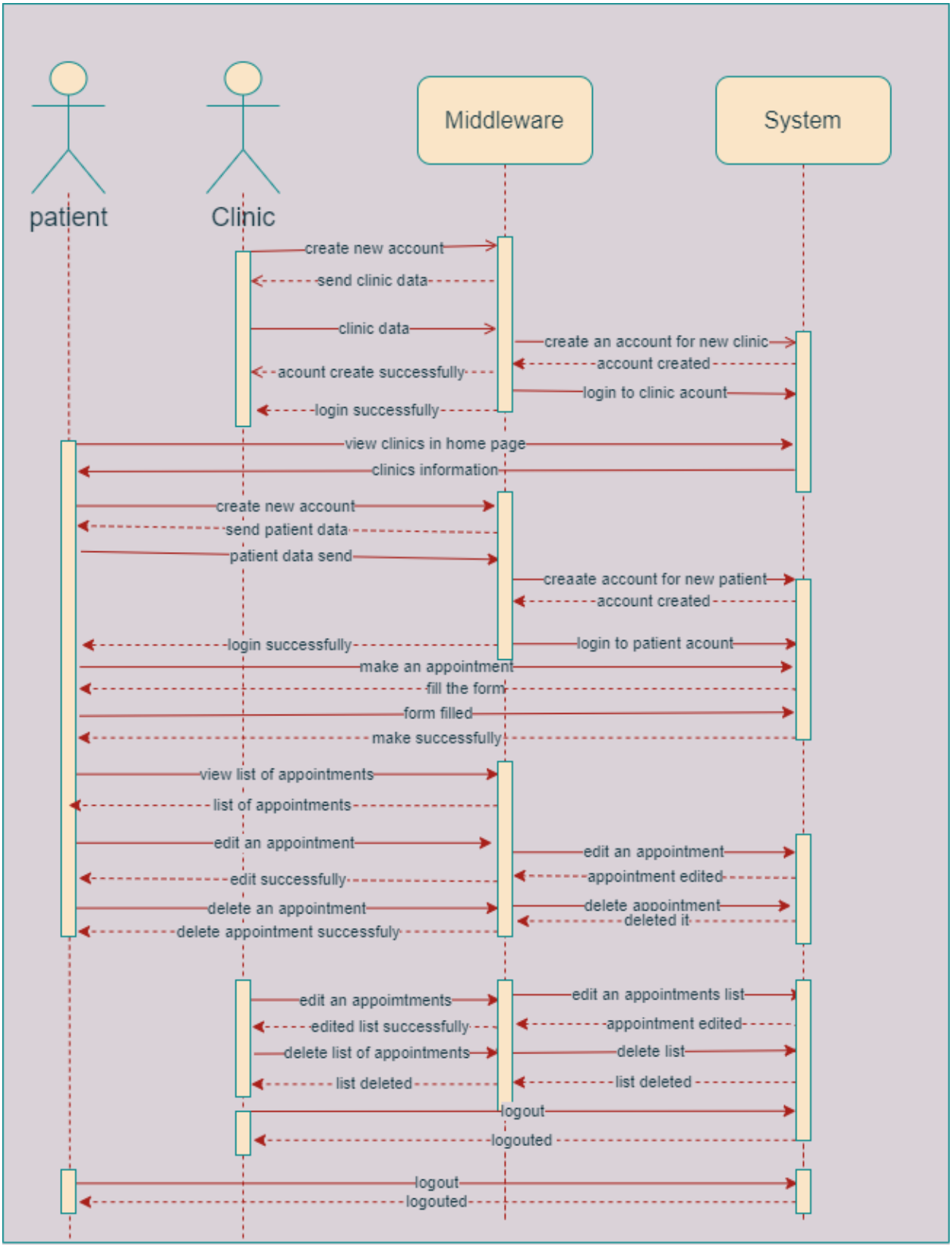


*Figure 2sequence diagram*

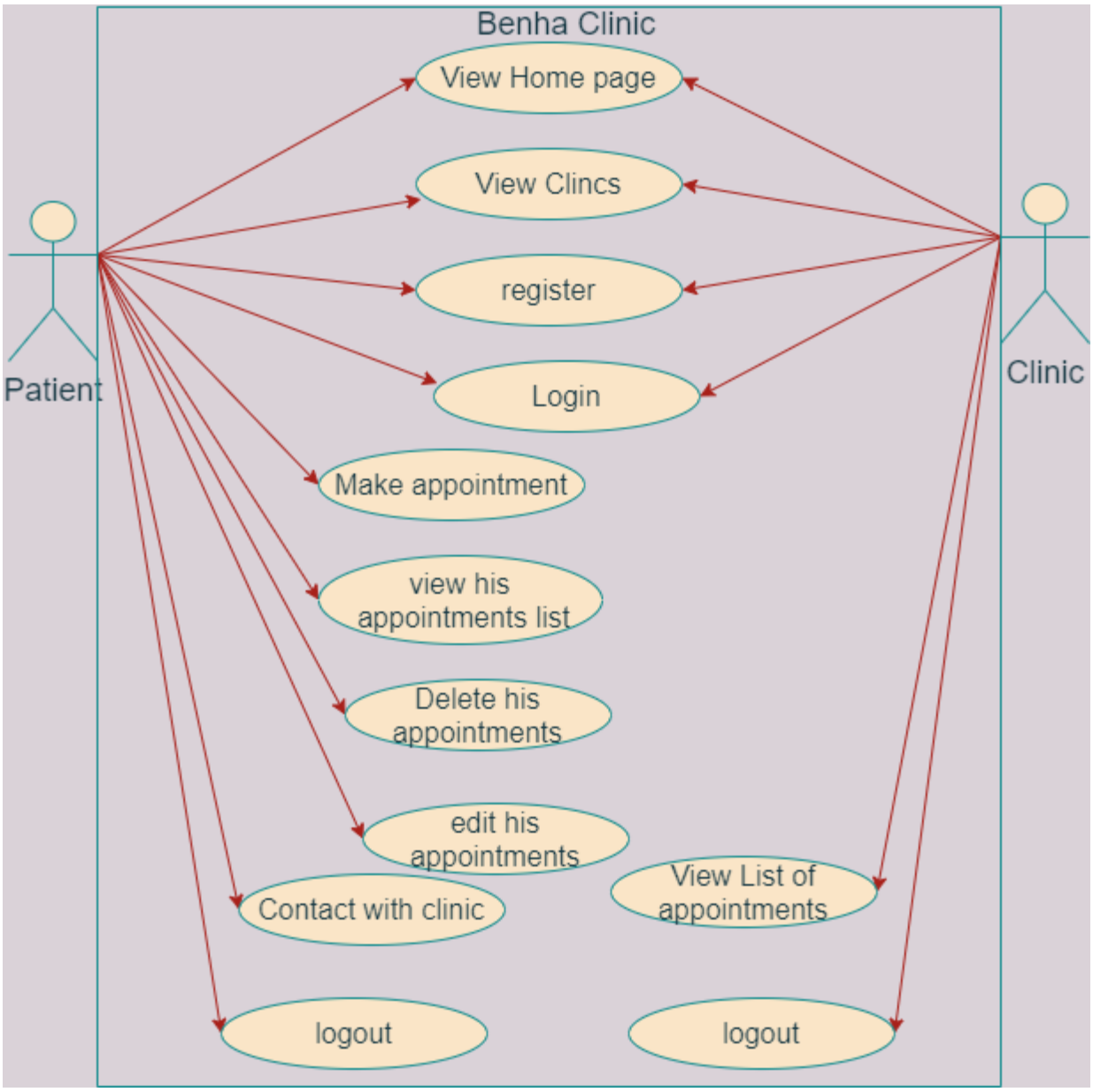


*Figure 3use case diagram*

System Forms and Reports Screens Using (Pencil):

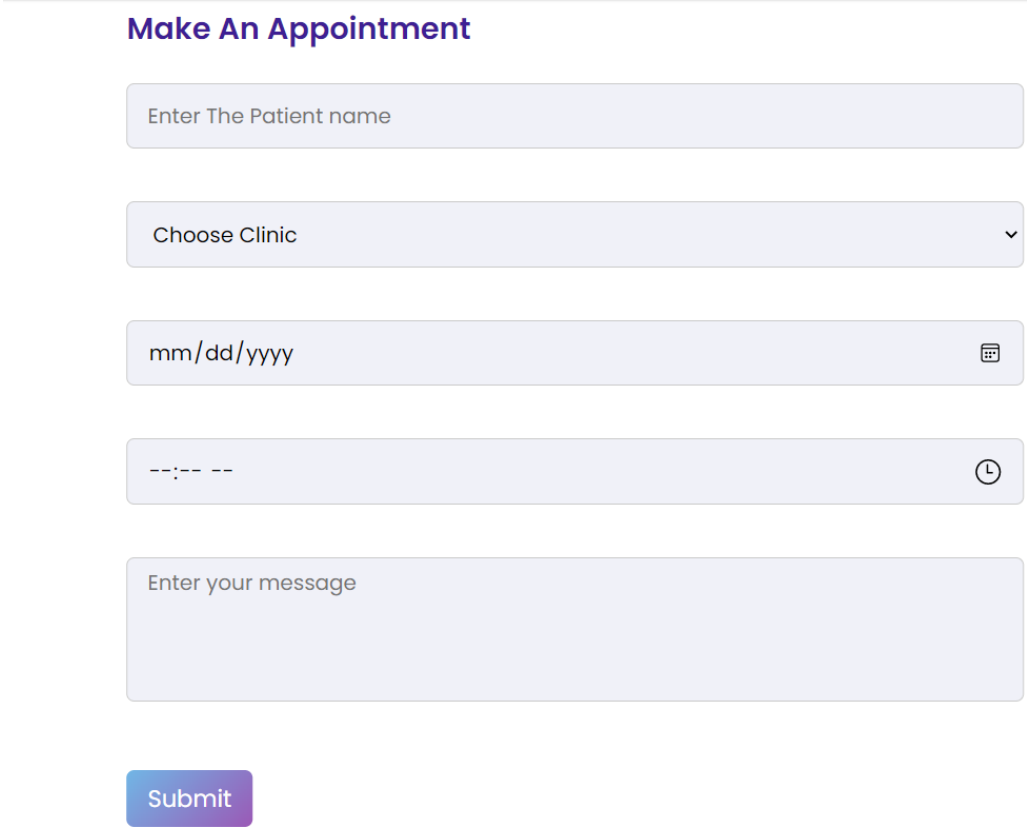


*Figure 4form make appointments*

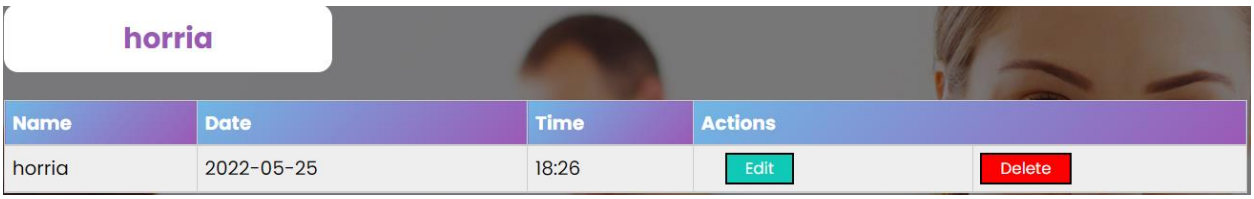


*Figure 5report patient appointments*

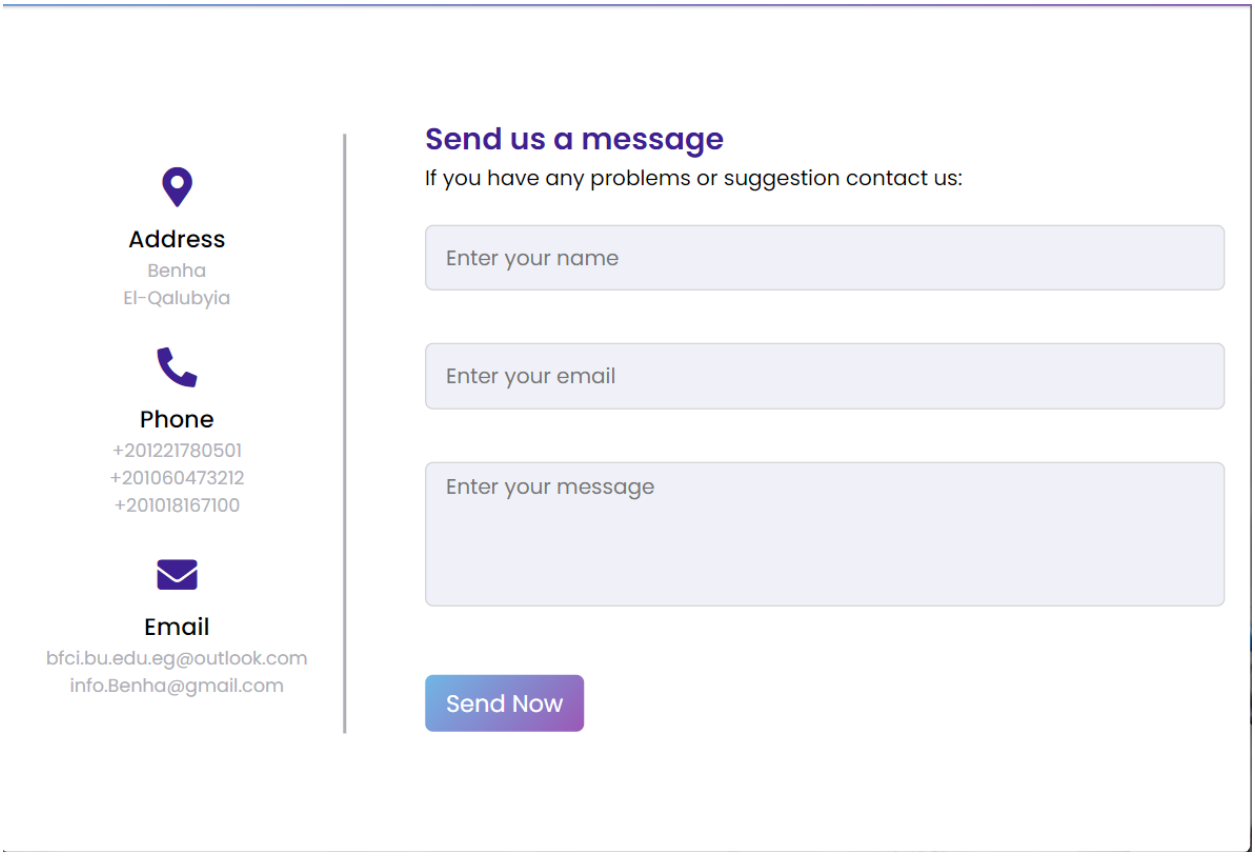


*Figure 6contact us form*

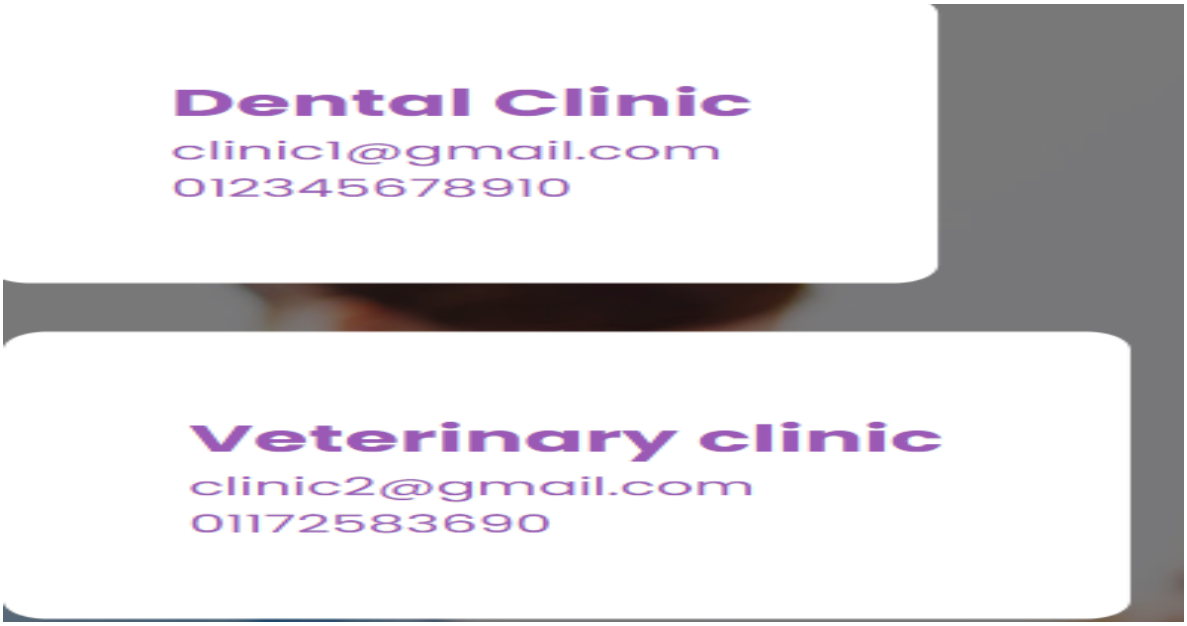


*Figure 7report of clinics in our system*


*Figure 8about our site*

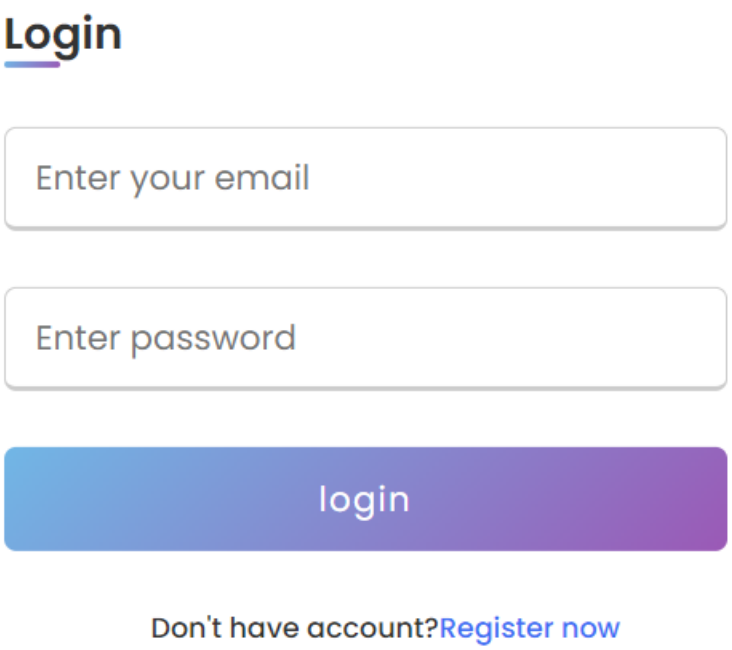


*Figure 9 form login for patients*

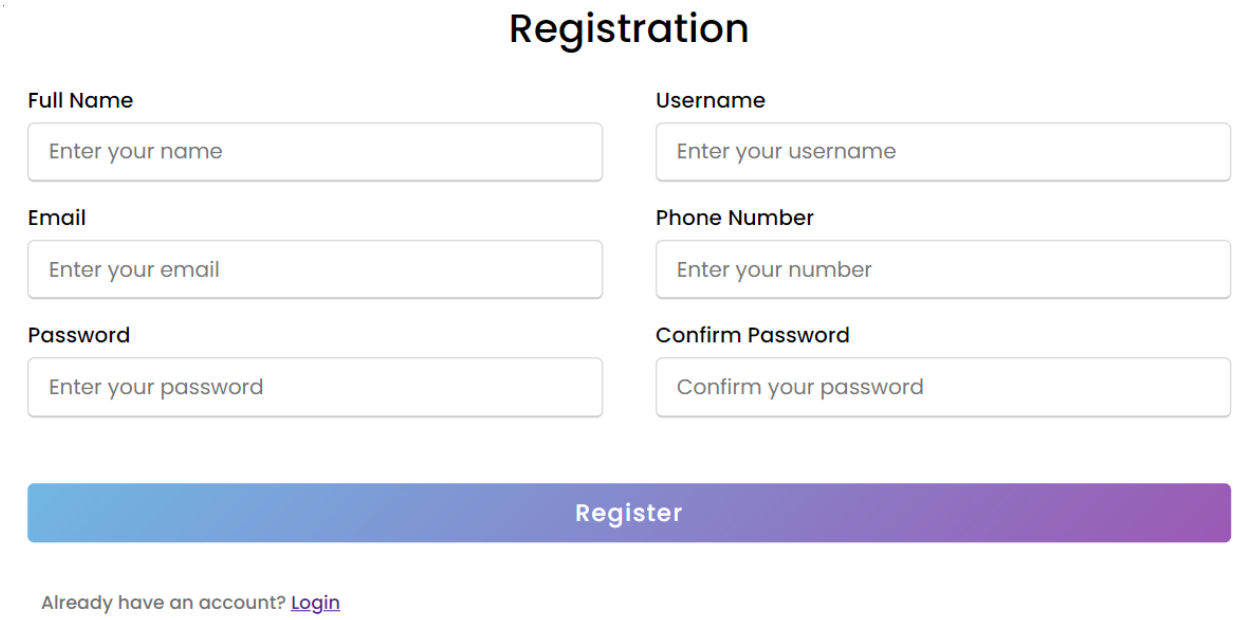


*Figure 11form registration for patients*

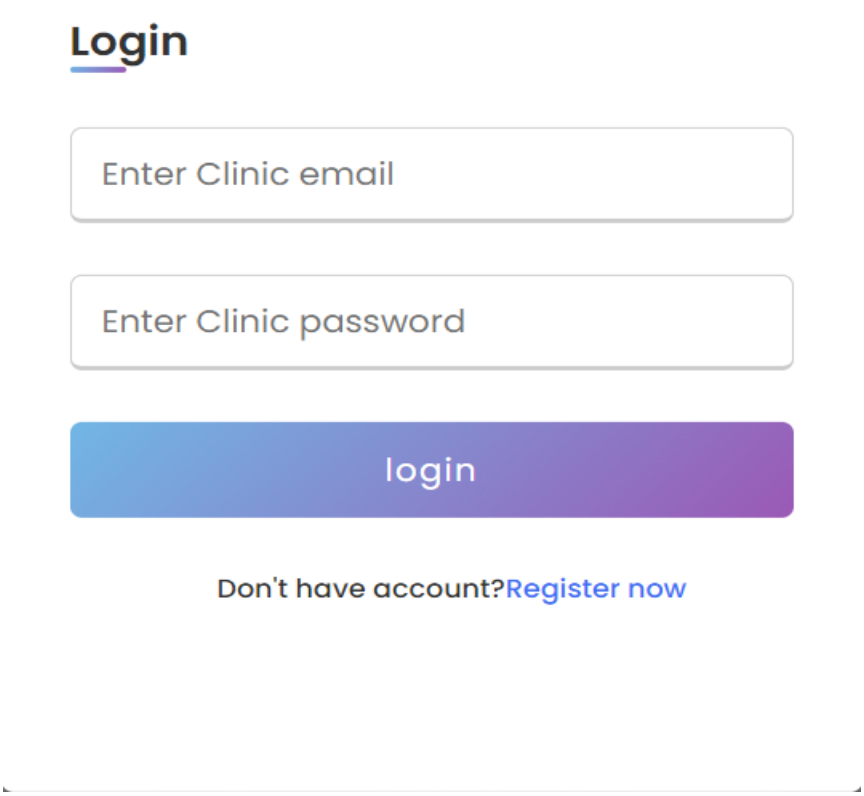


*Figure 10form login for patient*

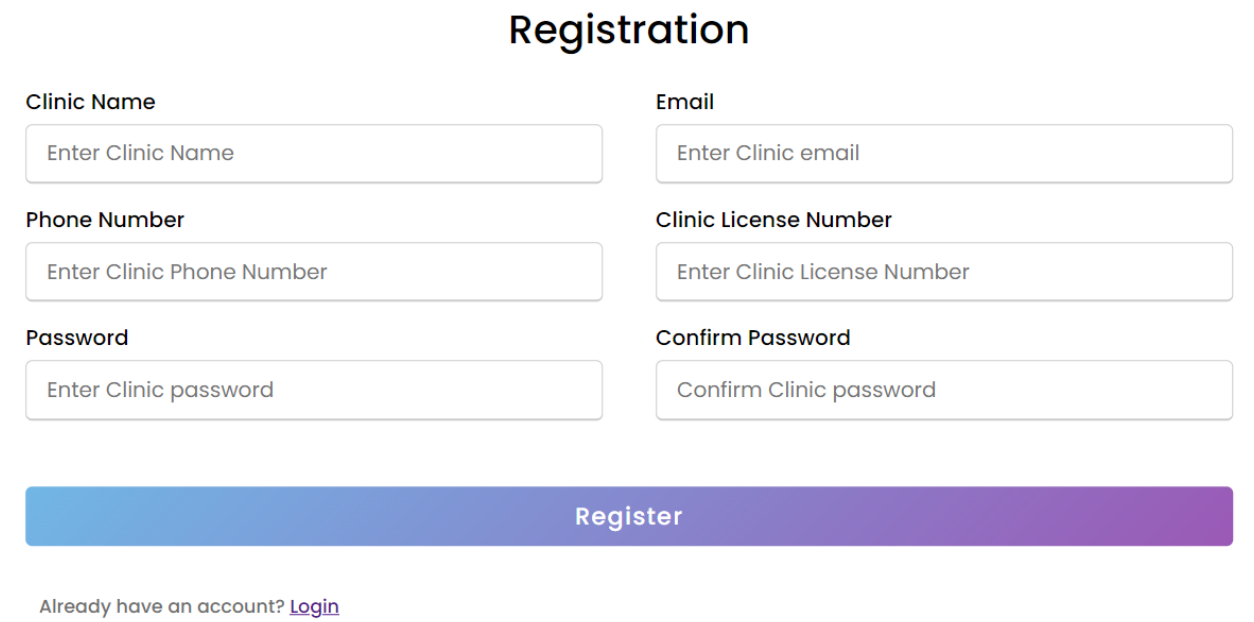


*Figure 12form registration for clinics*

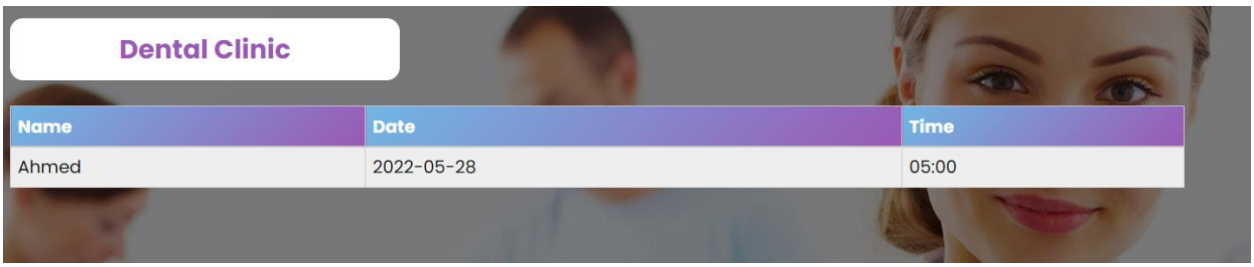


*Figure 13report clinic appointments1*

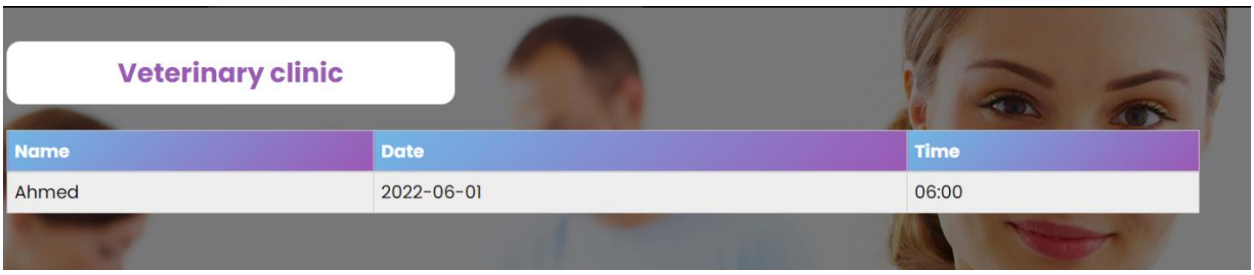


*Figure 14report clinic appointments2*

## Database schema and tables screen:

| Tables (10) | |
|---|---|
| books | CREATE TABLE "books" ("id" integer not null primary key autoincrement, "user_id" integer not null, "bname" varchar not null, "clinic_id" integer not null, "date" date not null, "time" time not null, "br |
| contacts | CREATE TABLE "contacts" ("id" integer not null primary key autoincrement, "coname" varchar not null, "coemail" varchar not null, "comessage" text not null, "created_at" datetime, "updated_at" dat |
| cregisters | CREATE TABLE "cregisters" ("id" integer not null primary key autoincrement, "cname" varchar not null, "cemail" varchar not null, "cphone" varchar not null, "cnum" varchar not null, "Password" varch |
| failed_jobs | CREATE TABLE "failed_jobs" ("id" integer not null primary key autoincrement, "uuid" varchar not null, "connection" text not null, "queue" text not null, "payload" text not null, "exception" text not nul |
| migrations | CREATE TABLE "migrations" ("id" integer not null primary key autoincrement, "migration" varchar not null, "batch" integer not null) |
| password_resets | CREATE TABLE "password_resets" ("email" varchar not null, "token" varchar not null, "created_at" datetime) |
| personal_access_tokens | CREATE TABLE "personal_access_tokens" ("id" integer not null primary key autoincrement, "tokenable_type" varchar not null, "tokenable_id" integer not null, "name" varchar not null, "token" varcha |
| pregisters | CREATE TABLE "pregisters" ("id" integer not null primary key autoincrement, "pname" varchar not null, "pusername" varchar not null, "pemail" varchar not null, "pphone" varchar not null, "password" |
| sqlite_sequence | CREATE TABLE sqlite_sequence(name,seq) |
| users | CREATE TABLE "users" ("id" integer not null primary key autoincrement, "name" varchar not null, "email" varchar not null, "email_verified_at" datetime, "password" varchar not null, "remember_toke |

*Figure 16database schema1*

null, "date" date not null, "time" time not null, "bmessage" text, "created_at" datetime, "updated_at" datetime, foreign key("user_id") references "Pregisters"("id") on delete cascade, foreign key("clinic_id") references "Cregisters"("id") on delete cascade)
t not null, "created_at" datetime, "updated_at" datetime)
ot null, "cnum" varchar not null, "Password" varchar not null, "Password_confirmation" varchar not null, "created_at" datetime, "updated_at" datetime)
ll, "payload" text not null, "exception" text not null, "failed_at" datetime default CURRENT_TIMESTAMP not null)

r not null, "name" varchar not null, "token" varchar not null, "abilities" text, "last_used_at" datetime, "created_at" datetime, "updated_at" datetime)
har not null, "pphone" varchar not null, "password" varchar not null, "password_confirmation" varchar not null, "created_at" datetime, "updated_at" datetime)

ime, "password" varchar not null, "remember_token" varchar, "created_at" datetime, "updated_at" datetime)

*Figure 15database schema2*

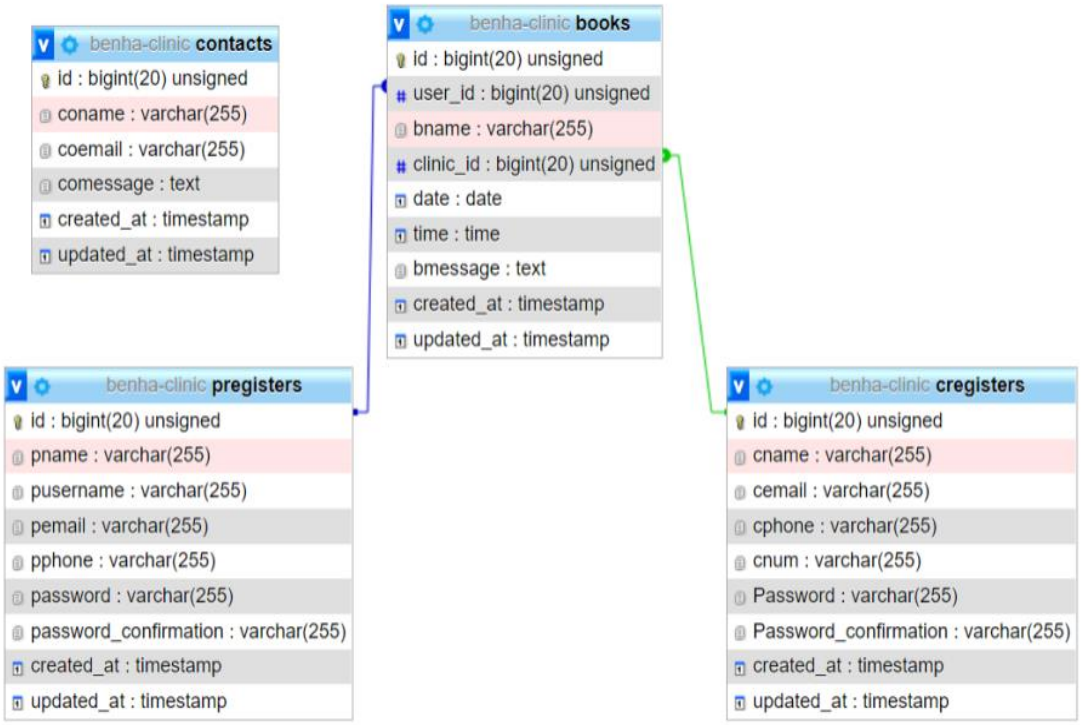


*Figure 17database schema design*

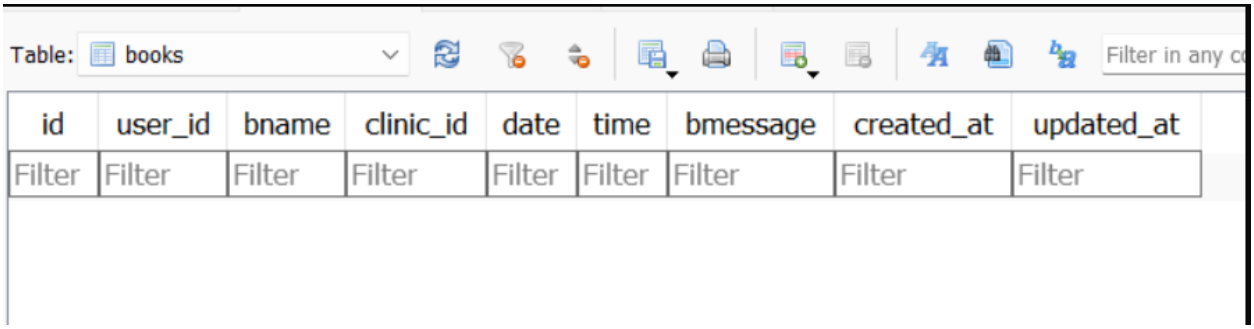


*Figure 18books database*

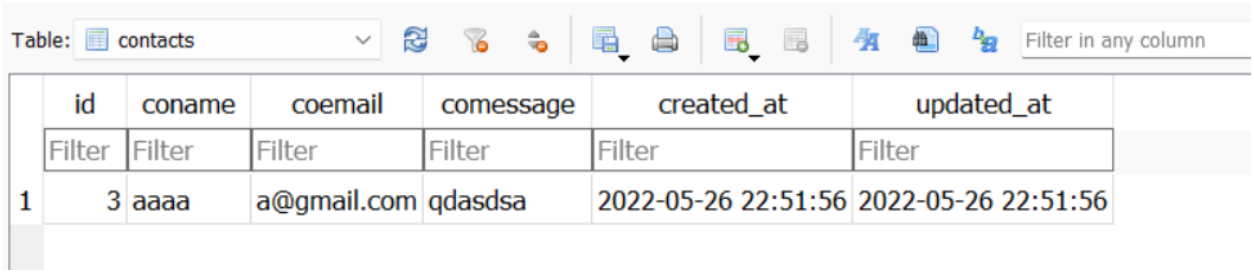


*Figure 19contact table*

Table: cregisters

Filter in any column

| id | cname | cemail | cphone | cnum | Password | Password_confirmation | created_at | updated_at |
|---|---|---|---|---|---|---|---|---|
| Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 Dental Clinic | clinic1@gmail.com | 012345678910 | 1234 | eyJpdiI6IlQyQTZpVVRzaHZVQ25We... | eyJpdiI6InUwNEZtVW1mdExiTUZGczl... | 2022-05-25 06:12:37 | 2022-05-25 06:12:37 |
| 2 | 2 Veterinary clinic | clinic2@gmail.com | 01172583690 | 12345 | eyJpdiI6IjVEL2Qzd0kxQ2xSM2xJcHR2... | eyJpdiI6Im04Ti9rYXplQm15QnoxYkR... | 2022-05-26 16:31:21 | 2022-05-26 16:31:21 |

*Figure 20registration table for clinics*

Table: pregisters

Filter in any column

| id | pname | pusername | pemail | pphone | password | password_confirmation | created_at | updated_at |
|---|---|---|---|---|---|---|---|---|
| Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 a | a | a@gmail.com | 0123456789 | eyJpdiI6IlBaVWxFTDQ3TVhyVGxhSE... | eyJpdiI6Imo4aHVlUVQrSDdSMG1YcU... | 2022-05-25 06:16:09 | 2022-05-25 06:16:09 |
| 2 | 2 b | bb | b@gmail.com | 0123456789 | eyJpdiI6IldzU1RhU0E4MGFGTHBqVm... | eyJpdiI6Ikhhcnk1b0hRZURqaDQ3aE5... | 2022-05-26 23:12:44 | 2022-05-26 23:12:44 |

*Figure 21registeration table or patient*

## proposed testing methods:

*Table 1test table*

| **Feature/Interface** | **Test case** | **Result** | **Type of error** |
|---|---|---|---|
| Registration. | Actors sign up process | Pass | None |
| Login | Actor sign-in process | Pass | None |

| | | | |
|---|---|---|---|
| Make an appointment | An actor can make an appointment | Pass | None |
| View actor appointments. | An actor can view their appointments. | Pass | None |
| Contact us | Actors can contact us. | Pass | None |
| Delete appointments | The patient can delete his appointments | Pass | None |
| Edit appointments | The patient can edit his appointments | Pass | None |
| View clinics | Actors can view clinics. | Pass | None |
| View information about us | Actors can view information about us | Pass | None |

| | | | |
|---|---|---|---|
| View appointment details | An actor can view appointment details. | Pass | None |

| | | | |
|---|---|---|---|
| View Home page | Actors can view the home page | Pass | None |
| Fast access to the user account from other pages | Actors can go back to their accounts from other pages | Pass | None |
| Logout | Actors can logout from their account | Pass | None |

*Table 2test database table*

| Verifying output data | Verifying incoming data |
|---|---|
| All the desired data was retrieved correctly from the database. | All the data entered into the database through the application was entered and stored in the database correctly. |

## Database Tables data:

# Tables (10)

| Name | Type | Schema |
|---|---|---|
| **books** | | CREATE TABLE "books" ("id" integer not null primary key autoincrement, "user_id" integer not null, "bname" varchar not null, "clinic_id" integer not null, "date" date not null, "time" time not null, "bmessage" text, "created_at" datetime, "updated_at" datetime, foreign key("user_id") references "Registers"("id") on delete cascade, foreign key("clinic_id") references "Cregisters"("id") on delete cascade) |
| id | integer | "id" integer NOT NULL |
| user_id | integer | "user_id" integer NOT NULL |
| bname | varchar | "bname" varchar NOT NULL |
| clinic_id | integer | "clinic_id" integer NOT NULL |
| date | date | "date" date NOT NULL |
| time | time | "time" time NOT NULL |
| bmessage | text | "bmessage" text |
| created_at | datetime | "created_at" datetime |
| updated_at | datetime | "updated_at" datetime |
| **contacts** | | CREATE TABLE "contacts" ("id" integer not null primary key autoincrement, "coname" varchar not null, "coemail" varchar not null, "comessage" text not null, "created_at" datetime, "updated_at" datetime) |
| id | integer | "id" integer NOT NULL |
| coname | varchar | "coname" varchar NOT NULL |
| coemail | varchar | "coemail" varchar NOT NULL |
| comessage | text | "comessage" text NOT NULL |
| created_at | datetime | "created_at" datetime |
| updated_at | datetime | "updated_at" datetime |
| **cregisters** | | CREATE TABLE "cregisters" ("id" integer not null primary key autoincrement, "cname" varchar not null, "cemail" varchar not null, "cphone" varchar not null, "cnum" varchar not null, "Password" varchar not null, "Password_confirmation" varchar not null, "created_at" datetime, "updated_at" datetime) |
| id | integer | "id" integer NOT NULL |
| cname | varchar | "cname" varchar NOT NULL |
| cemail | varchar | "cemail" varchar NOT NULL |
| cphone | varchar | "cphone" varchar NOT NULL |
| cnum | varchar | "cnum" varchar NOT NULL |
| Password | varchar | "Password" varchar NOT NULL |
| Password_confirmation | varchar | "Password_confirmation" varchar NOT NULL |
| created_at | datetime | "created_at" datetime |
| updated_at | datetime | "updated_at" datetime |
| **failed_jobs** | | CREATE TABLE "failed_jobs" ("id" integer not null primary key autoincrement, "uuid" varchar not null, "connection" text not null, "queue" text not null, "payload" text not null, "exception" text not null, "failed_at" datetime default CURRENT_TIMESTAMP not null) |
| id | integer | "id" integer NOT NULL |
| uuid | varchar | "uuid" varchar NOT NULL |
| connection | text | "connection" text NOT NULL |
| queue | text | "queue" text NOT NULL |
| payload | text | "payload" text NOT NULL |
| exception | text | "exception" text NOT NULL |
| failed_at | datetime | "failed_at" datetime NOT NULL DEFAULT CURRENT_TIMESTAMP |

1

*Figure 22database tables data*

| Name | Type | Schema |
|---|---|---|
| **migrations** | | CREATE TABLE "migrations" ("id" integer not null primary key autoincrement, "migration" varchar not null, "batch" integer not null) |
| id | integer | "id" integer NOT NULL |
| migration | varchar | "migration" varchar NOT NULL |
| batch | integer | "batch" integer NOT NULL |
| **password_resets** | | CREATE TABLE "password_resets" ("email" varchar not null, "token" varchar not null, "created_at" datetime) |
| email | varchar | "email" varchar NOT NULL |
| token | varchar | "token" varchar NOT NULL |
| created_at | datetime | "created_at" datetime |
| **personal_access_tokens** | | CREATE TABLE "personal_access_tokens" ("id" integer not null primary key autoincrement, "tokenable_type" varchar not null, "tokenable_id" integer not null, "name" varchar not null, "token" varchar not null, "abilities" text, "last_used_at" datetime, "created_at" datetime, "updated_at" datetime) |
| id | integer | "id" integer NOT NULL |
| tokenable_type | varchar | "tokenable_type" varchar NOT NULL |
| tokenable_id | integer | "tokenable_id" integer NOT NULL |
| name | varchar | "name" varchar NOT NULL |
| token | varchar | "token" varchar NOT NULL |
| abilities | text | "abilities" text |
| last_used_at | datetime | "last_used_at" datetime |
| created_at | datetime | "created_at" datetime |
| updated_at | datetime | "updated_at" datetime |
| **pregisters** | | CREATE TABLE "pregisters" ("id" integer not null primary key autoincrement, "pname" varchar not null, "pusername" varchar not null, "pemail" varchar not null, "pphone" varchar not null, "password" varchar not null, "password_confirmation" varchar not null, "created_at" datetime, "updated_at" datetime) |
| id | integer | "id" integer NOT NULL |
| pname | varchar | "pname" varchar NOT NULL |
| pusername | varchar | "pusername" varchar NOT NULL |
| pemail | varchar | "pemail" varchar NOT NULL |
| pphone | varchar | "pphone" varchar NOT NULL |
| password | varchar | "password" varchar NOT NULL |
| password_confirmation | varchar | "password_confirmation" varchar NOT NULL |
| created_at | datetime | "created_at" datetime |
| updated_at | datetime | "updated_at" datetime |
| **sqlite_sequence** | | CREATE TABLE sqlite_sequence(name,seq) |
| name | | "name" |
| seq | | "seq" |
| **users** | | CREATE TABLE "users" ("id" integer not null primary key autoincrement, "name" varchar not null, "email" varchar not null, "email_verified_at" datetime, "password" varchar not null, "remember_token" varchar, "created_at" datetime, "updated_at" datetime) |
| id | integer | "id" integer NOT NULL |
| name | varchar | "name" varchar NOT NULL |
| email | varchar | "email" varchar NOT NULL |
| email_verified_at | datetime | "email_verified_at" datetime |
| password | varchar | "password" varchar NOT NULL |



*Figure 23database tables data2*

| Name | Type | Schema |
|---|---|---|
| remember_token | varchar | "remember_token" varchar |
| created_at | datetime | "created_at" datetime |
| updated_at | datetime | "updated_at" datetime |

## Indices (5)

| Name | Type | Schema |
|---|---|---|
| **failed_jobs_uuid_unique** | | CREATE UNIQUE INDEX "failed_jobs_uuid_unique" on "failed_jobs" ("uuid") |
| uuid | | "uuid" |
| **password_resets_email_index** | | CREATE INDEX "password_resets_email_index" on "password_resets" ("email") |
| email | | "email" |
| **personal_access_tokens_token_unique** | | CREATE UNIQUE INDEX "personal_access_tokens_token_unique" on "personal_access_tokens" ("token") |
| token | | "token" |
| **personal_access_tokens_tokenable_type_tokenable_id_index** | | CREATE INDEX "personal_access_tokens_tokenable_type_tokenable_id_index" on "personal_access_tokens" ("tokenable_type", "tokenable_id") |
| tokenable_type | | "tokenable_type" |
| tokenable_id | | "tokenable_id" |
| **users_email_unique** | | CREATE UNIQUE INDEX "users_email_unique" on "users" ("email") |
| email | | "email" |

## Views (0)

| Name | Type | Schema |
|---|---|---|

## Triggers (0)

| Name | Type | Schema |
|---|---|---|

*Figure 24database tables data3*

## Conclusion

The problem is the Clinics were very crowded, so we came up with the solution.

the patient now can book his appointment online and come to the clinic during his appointment, on the other hand, the clinic can be registered on the site and show all the appointments. First, we made a site to serve one clinic and its patients, then we generalized the idea. The site can serve more than one clinic and many patients.

## Resources:


1. Jazayeri, M. (2004, September). The education of a software engineer. In Proceedings. 19th International Conference on Automated Software Engineering, 2004. (pp. xviii-xxvii). IEEE.

2. Li, P. L., Ko, A. J., & Zhu, J. (2015, May). What makes a great software engineer?. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (Vol. 1, pp. 700- 710). IEEE.

3. Meade, E., O'Keeffe, E., Lyons, N., Lynch, D., Yilmaz, M., Gulec, U., ... & Clarke, P. M. (2019, September). The changing role of the software engineer. In European Conference on Software Process Improvement (pp. 682-694). Springer, Cham.

4. Baxter, R., Hong, N. C., Gorissen, D., Hetherington, J., & Todorov, I. (2012, September). The research software engineer. In Digital Research Conference, Oxford (pp. 1-3).

5. Wirth, N. (1981). Lilith: A personal computer for the software engineer. In Microcomputer System Design (pp. 349-397). Springer, Berlin, Heidelberg.