

***Full-Stack Project Revere Dental Clinic***

Tectigon Academy

30 Dhjetor 2024

DRILON ZOGAJ

**Hyrje**

Ky dokument është shkruar me qëllim dokumentimin e ëeb aplikacionit të quajtur Revere Clinic.

Ky projekt është një aplikacion i ndërtuar duke përdorur teknologjitë më të fundit full stack, që

ka për qëllim të ofrojë një zgjidhje funksionale për një kategori përdoruesish (pacient dhe mjek

dentar). Qëllimi kryesor i këtij projekti është lehtësimi i procesit të rezervimeve dhe përmirësimi

i menaxhimit të termineve nga ana e mjekut duke ofruar një dizajn

User-Friendly të thjeshtë dhe praktik për ta përdorur.

## Përshkrimi i sistemit

Stakeholders:

Dy stakeholders kryesorë të këtij projekti janë:

* Drilon Zogaj – Full Stack Developer
* Tectigon Academy – Client

Koha e zhvillimit:

Projekti ka filluar më datë 04/12/2024.

Pjesa e frontend-it ka filluar, më datë 04/12/2024 dhe ka përfunduar, më datë 16/12/2024

Pjesa e backend-it ka filluar, më datë 17/12/2024 dhe ka pëfunduar, më datë 27/12/2024, njëtrajtësisht është zhvilluar edhe databaza e projektit.

Ky projekt është në fazen përfundimtare, projekti ka përfunduar, më datë 27/12/2024 dhe do të dorëzohet, më datë 30/12/2024

Teknologjitë e përdorura:

Pjesa Front-End (Pjesa e Ndërfaqes së Përdoruesit):

* HTML: Gjuhë e markup-it që përdoret për të strukturuar përmbajtjen e faqeve web.
* CSS: Përdoret për stilizimin e aplikacionit dhe dizajnit të ndërfaqes, duke siguruar që aplikacioni të duket dhe të funksionojë bukur në çdo pajisje.
* JavaScript: Gjuhë programimi që mundëson sjelljen dinamike të aplikacionit, duke mundësuar veprime të ndryshme të ndërfaqes, si animacione dhe interaktivitet.
* ReactJS: Një library e JavaScript për ndërtimin e ndërfaqeve të përdoruesit (UI). Me React, aplikacionet bëhen më modulare dhe të menaxhueshme përmes komponenteve të ripërdorshëm.
* Vite: Një mjet ndërtimi modern që ofron një përvojë më të shpejtë dhe më efikase zhvillimi. Vite mundëson përditësime të shpejta të moduleve dhe krijimin e ndërtimit optimizuar për prodhim.

Pjesa e Back-End (Pjesa e Serverit):

* PHP: Një gjuhë skenari serveri që përdoret për menaxhimin e logjikës së biznesit dhe krijimin e pikave të API (të cilat mundësojnë komunikimin mes front-end dhe back-end).
* PDO (PHP Data Objects): Një ndërfaqe e sigurt dhe fleksibël për lidhjen dhe pyetjen e bazave të të dhënave MySQL. PDO përdor deklarata të përgatitura për të parandaluar sulmet SQL injection dhe për të siguruar pyetje të sigurta.
* MySQL: Një sistem menaxhimi të bazës së të dhënave relacional që përdoret për të ruajtur të dhënat e strukturuara të aplikacionit, si informacionet e përdoruesve, produktet, dhe të dhëna të tjera.
* https://github.com/DrilonZogaj/Revere

## Kërkesat e sistemit

Kërkesat funksionale:

1. Sistemi duhet të lejon pacienti të aplikoj për termin në klinikë.
2. Sistemi nuk duhët të lejoj pacientin të aplikoj për terminin e njejtë më shumë se një (1) herë.
3. Sistemi duhet të lejon pacienti të kontaktoj kliniken.
4. Sistemi duhet të informoj pacienti që termini është krijuar me sukses.
5. Sistemi duhet të informoj pacienti që krijimi i terminit ka dështuar.
6. Sistemi duhet të informoj pacienti që mesazhi për kontakt me klinikë është realizuar me sukses.
7. Sistemi duhet të informoj pacienti që mesazhi për kontakt me klinikë nuk është realizuar me sukses.
8. Sistemi duhet të lejoj doktorin të pranoj terminin
9. Sistemi duhet të lejoj doktorin të mos e pranoj terminin.
10. Sistemi duhet të lejoj doktorin të fshij terminin.
11. Sistemi duhet të lejoj doktorin të krijoj termin nga të dhënat e derguara nga pacienti përmes konakt formes.
12. Sistemi duhet të lejoj doktorin të ndryshoj të dhënat e terminit.

Kërkesat jo funksionale:

1. Sistemi duhet të përpunojë 95% të kërkesave brenda 2 sekondave.
2. Ndërfaqja duhet të jetë User-Friendly dhe të mbështesë përdoruesit e rinjë.
3. Sistemi duhet të mbështesë rritjen nga 1000 në 10,000 përdorues aktivë në muaj pa ndonjë ndikim të dukshëm në performancë.
4. Sistemi duhet të jetë i aksesueshëm 24/7, me një nivel disponueshmërie prej 99.9%.
5. Sistemi duhet të jetë kompatibil me shfletuesit kryesorë (Chrome, Firefox, Edge, Safari) dhe pajisjet mobile.
6. Sistemi duhet të jetë deterministik që d.m.th për të njejten hyerje të jap të njejtin dalje.
7. Të dhënat e ruajtura në bazën e të dhënave duhet të jenë të disponueshme dhe të mund të aksesohen në kohë reale pa vonesa të mëdha.

## Arkitekura e Sistemit

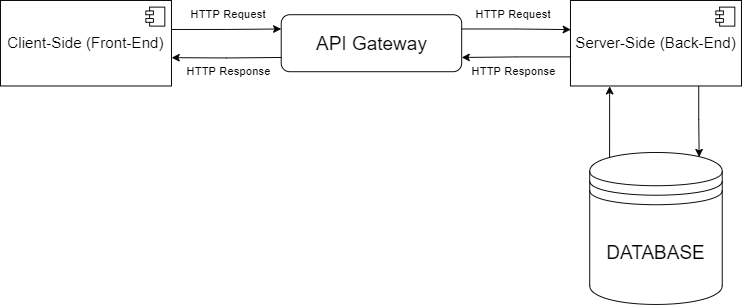


Figure . Diagrami i arkitektures se sistemit ne pika te trasha

Pershkrimi i komponënteve:

Client-Side (Front-End): Paraqet komponentin e front-endit apo pjesës që e sheh end-useri.

Server-Side (Back-End): Paraqet komponentin e back-endit apo pjesën e logjikës ku procesohen kërkesat (handle the requests) e end-userit.

Database: Paraqet databazen e sistemit në të cilën ruhet të gjitha të dhënat e sistemit.

API Gateway: Është mënyra e komunikimit të front-end me back-end. Kjo arrihet përmes HTTP Requests dhe HTTP Response qe mundësohen nga API i sistemit.

## Dizajni i databazes

Skema e databazes dhe ER-Diagrami

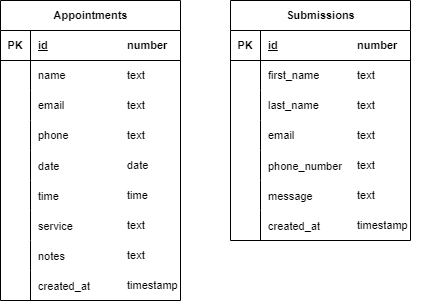


Figure 2. Database tables schema.

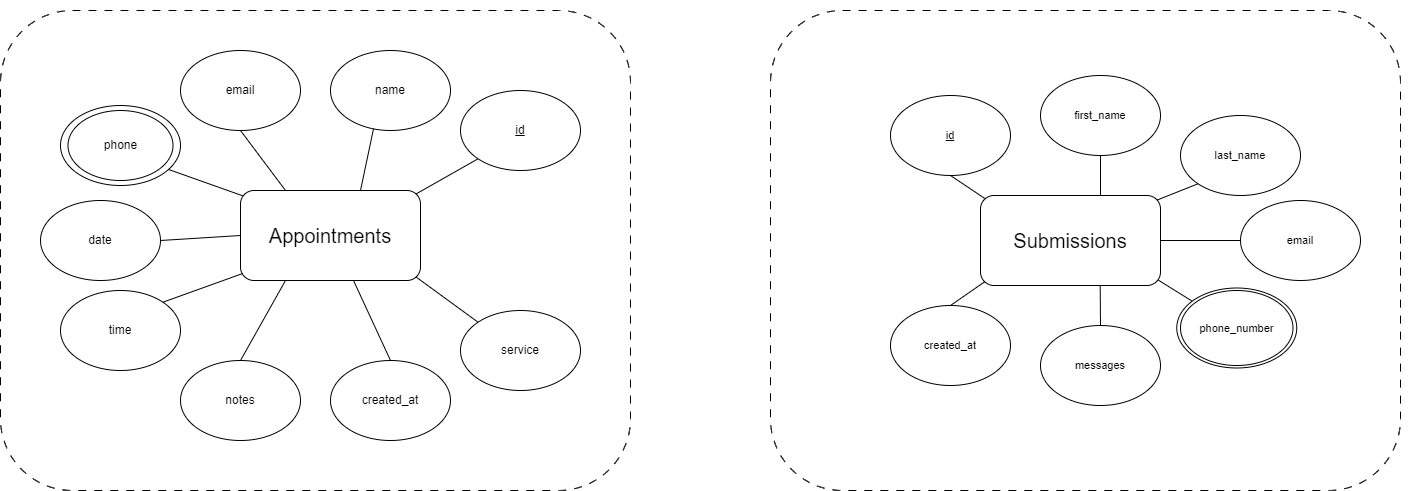


Figure 3 ER Diagrami.

Tabelat

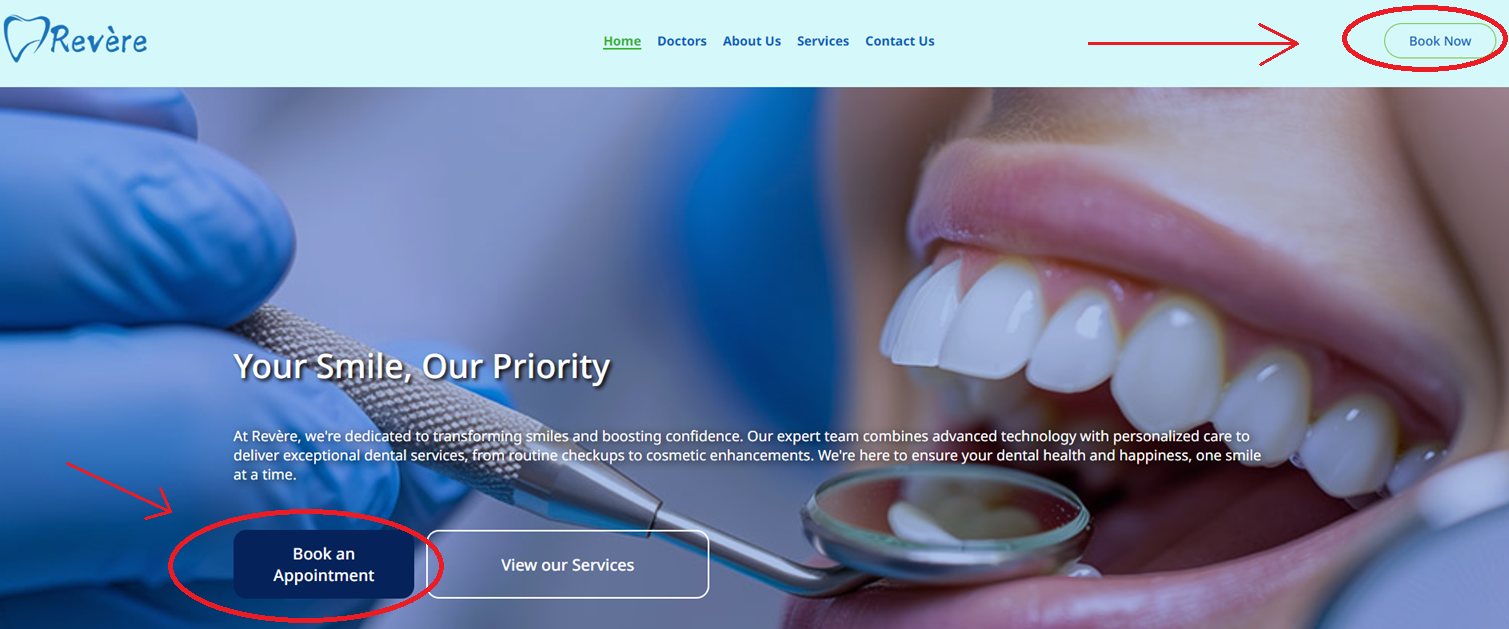
* Appointments: Është tabela që i ruan të dhënat e nevojshme të termineve si psh. Id e terminit, emri i plote i pacientit, numri i telefonit të pacientit, data e terminit për pacientin e caktuar, koha e terminit, sherbimin e zgjedhur dhe mesazhin.
* Submissions: Është tabela që i ruan të dhënat e nevojshme të mesazheve që pacienti i dërgon klinikës si psh. Id e mesazhit, emri i derguesit, mbiemri i derguesit, emaila, numri i telefonit, mesazhi qe ka derguar dhe kohen e dergimit.

Model Demonstrues

Rezervimi i Termineve (Book an Appointment)

1 Në bazë të shigjetave të kuqe të paraqitura në modelin demonstrues:

* Shigjeta e sipërme e djathtë tregon një buton me tekstin "Book Now", që duket në krye të faqes. Ky buton shërben për të rezervuar një takim drejtpërdrejt.
* Shigjeta e poshtme e majtë tregon butonin tjetër me tekstin "Book an Appointment", që gjendet në seksionin kryesor të faqes. Ky buton, me dizajn të dukshëm dhe të dallueshëm, është i vendosur për të tërhequr vëmendjen e përdoruesve drejt rezervimit të një takimi nga faqja hyrëse.

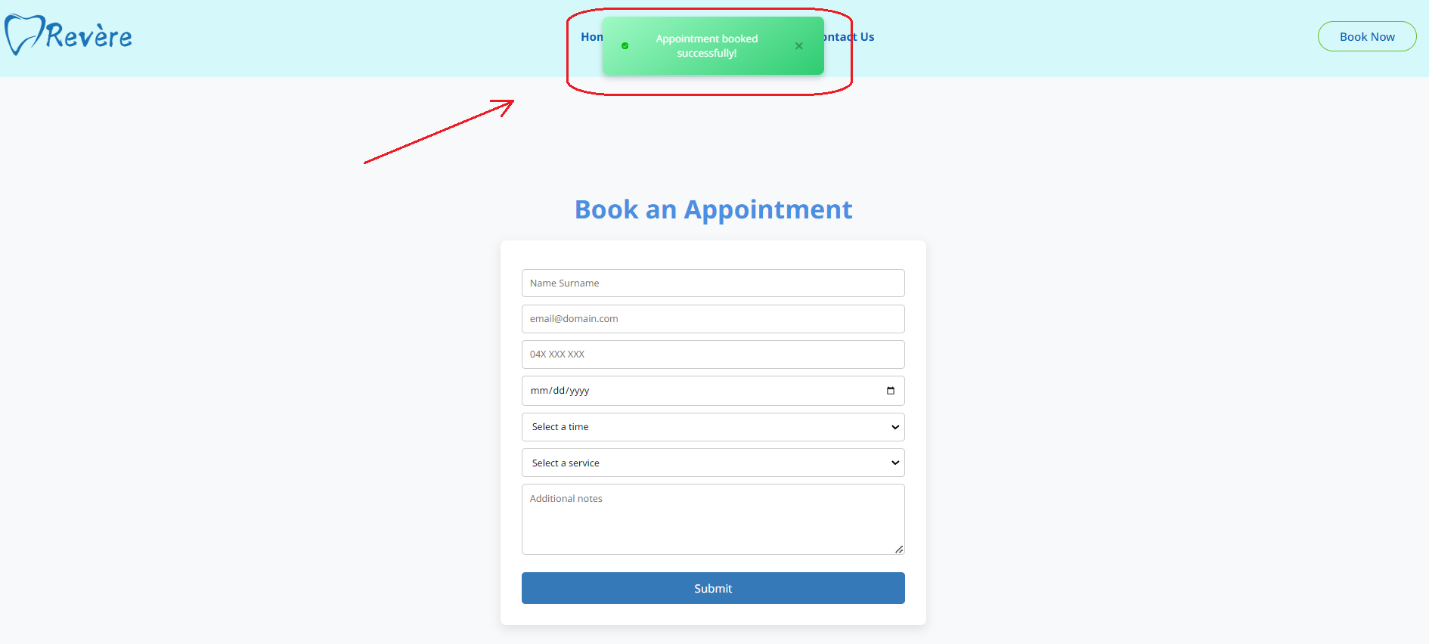


1.2 Pas klikimit në njërin nga butonate paraprak, perdouresit I shfaqet forma për rezervimin e një takimi, e cila përmban disa fusha për plotësim dhe një buton për dërgimin e të dhënave.

A screenshot of a computer

Description automatically generated

1.3 Kur përfundon plotësimi i formës nga përdouresi (pacienti) me të dhënat e nevojshme, shtypet butoni “Submit”, dhe shfaqet një njoftim konfirmues në krye të faqes :



2 Shigjetat dhe qarqet në foto theksojnë elemente të rëndësishme të faqes së "Contact Us" dhe kanë funksion të identifikojnë seksionet ose butonat kryesorë. Ja se çfarë tregojnë:

* + Shigjeta e kuqe në krye (te menuja): Tregon se përdoruesi është në seksionin "Contact Us" (Na kontaktoni). Kjo është një shenjë vizuale për të treguar se cili seksion është aktiv.
  + Qarku i kuq rreth butonit "Submit": Thekson butonin "Submit", që përdoret për të dërguar informacionin e plotësuar në formën e kontaktit. Ky është hapi i fundit që përdoruesi duhet të ndjekë për të përfunduar procesin.

A screenshot of a contact us

Description automatically generated

* + Pas plotësimit të formës së kontaktit të dhënat ruhen në database, te njëjtat forma të plotësuara mund të menaxhohen nga doktorri kryesorë (admin), të cilat perms API shfaqen në dashboard:
  + URL: localhost:5173/messages

A screenshot of a computer

Description automatically generated

* + Admin mund të realizojë një rezervim të ri permës butonit në kontaktet e derguara. Duke klikuar në butonin “Add Appointment” drejtohet në faqen ku plotësohen të dhënat për termin:
  + URL: localhost:5173/appointments

A screenshot of a computer

Description automatically generated

Admin Log in

* + Admin ka mundësinë të kyqet në faqen ku shfaqen terminet dhe mesazhet e dërguara përmes URL : localhost:5173/admin-login
  + Të dhënat për qasje: Username : admin, Password: password.

A screenshot of a computer

Description automatically generated

Admin Dashboard

Pas kyqjes në sistemin për menaxhimin me rolin admin, shfaqen terminet nga databaza:

A screenshot of a computer

Description automatically generated

API Design

1. Add Appointment

* Endpoint: http://localhost/backend/db/addAppointments.php
* Method: POST
* Description: Adds a new appointment to the database.
* Request Payload:

json

{

"name": "string", // Emri i plotë i pacientit

"email": "string", // Email addressa

"phone": "string", // Numri i telefonit

"date": "YYYY-MM-DD", // Data e rezervimit të terminit

"time": "HH:MM AM/PM", // Koha e terminit

"service": "string", // Shërbimi dentar

"notes": "string" // Shënime shtesë

}

* Response:
  + Success:

json

{ “status": "success", "message": "Appointment created successfully!" }

* Failure:

Json

{"status": "error", "message": "Error creating appointment.”}

2. Get Appointments

* Endpoint: http://localhost/backend/db/getAppointments.php
* Method: GET
* Description: Rikthen listën e të gjitha termineve nga databaza.
* Request Payload: None
* Response:
* Success:

Json

[

{

"appointment\_id": "number",

"name": "string",

"email": "string",

"phone": "string",

"date": "YYYY-MM-DD",

"time": "HH:MM AM/PM",

"service": "string",

"notes": "string"

},

...

]

* Failure:

{ "status": "error", "message": "Error fetching appointments." }

3. Delete Appointment

* Endpoint: http://localhost/backend/db/deleteAppointments.php
* Method: DELETE
* Description: Fshin një takim nga databaza bazuar në ID-në e tij..
* Request Payload:

Json

{ "appointment\_id": "number" }

Response:

* Success:

json

{"status": "success”, “message": "Appointment deleted successfully!”}

* + Failure:

Json

{"status": "error”, “message": "Failed to delete appointment.”}

4. Update Appointment

* Pika e aksesit (Endpoint): <http://localhost/backend/db/updateAppointments.php>
* Metoda: POST
* Përshkrimi: Përditëson detajet e një termini ekzistues në databazë.
* Përmbajtja e kërkesës (Request Payload):

Json

{

"appointment\_id": "number",

"name": "string",

"email": "string",

"phone": "string",

"date": "YYYY-MM-DD",

"time": "HH:MM AM/PM",

"service": "string",

"notes": "string"

}

Response:

* Success:

json

{ "status": "success", "message": "Appointment updated successfully!" }

* + Failure:

{"status": "error", "message": "Failed to update appointment." }

Karakteristikat Kryesore

Autentikimi (Vetëm për Administratorët):

* Qasja e administratorit kontrollohet duke përdorur localStorage (adminAuthenticated).
* Përdoruesit që nuk janë të autentikuar ridrejtohen te faqja e hyrjes.

CRUD Operations:

* + Create Shto një termin të ri përmes formularit të rezervimit..
  + Read: Fetch all terminet përmes panelit kryesor (dashboard).
  + Update: Përditëso detajet e termineve.
  + Delete: Fshi terminet si admin.

Validimi:

* Validimi i të dhënave në frontend siguron integritetin e të dhënave para se të kryhen thirrjet në API.

Trajtimi i Gabimeve(Error Handling):

* Përgjigjet e API përfshijnë mesazhe gabimi për diagnostikim më të mirë dhe kthim informativ për përdoruesin.

Procesi i zhvillimit

Veglat:

* Visual Studio Code (VS Code): Është vegla e përdorur për të shkruar logjiken e kodit për sistem.
* GitHub: Është përdorur për të hostuar kodin online.

Metodologjia:

Metodologjia e përdorur për të zhvilluar aplikacionin është “Incremental Microservices Development”. Metodologji kjo që zhvillon projektin në pjesë të vogla te quajtura mikroservice, të cilat më pas me mikroservice të tjera të integruara dhe ndërlidhura me njëra tjetren e formojnë aplikacionin. Kjo metodë e bënë më të lehtë mirëmbajtjen e aplikacionit.

Testimet

Testimet e bëra:

* Unit Testing
* Integration Testing
* End-to-End Testing
* Smoke Testing

Më poshtë gjeni dy test case të ndryshëm për dy raste të ndryshme.

### Test Case: Rezervo terminin

#### Test Case ID: TC001

#### Test Case Title: Rezervimi i terminit përmes formës për rezervime.

Preconditions:

1. Përdoruesi duhet të ketë qasje në web aplikacion.
2. Rezervo terminin “Book an Appointment” duhet të jetë e çasshme.
3. Koha e rezervimit duhet të jetë e lirë.

Hapat e testimit:

1. Hap faqen për të bërë termine:
   * Action: Kliko ne buttonin “Book an Appointment” ose “Book noë”
   * Rezultati i pritur: Faqja për të bërë terminet hapet me sukses.
2. Select Appointment Details:
   * Action: Mbush të dhënat e kërkuara
     + Emri i plote: Shkruan emrin e pacientit.
     + Email-a: Email addressa e pacientint.
     + Numri i telefonit: Shenon numrin e telefonit të pacientit.
     + Data: Zgjedh datën e terminit.
     + Koha: Zgjedhë kohen e terminit.
     + Sherbimi: Zgjedh sherbimin e deshiruar.
     + Mesazhi (Jo e domosdoshme): Mesazhi për klinikën (doktorin)
   * Rezultatet e pritura: Termini realizohet me sukses
3. Dërgo kërkesën për termin:
   * Action: Kliko buttonin "Submit".
   * Rezultatet e pritura: Shfaqet një alert që tregon që termini është realizuar me sukses.

Postconditions

1. Të dhënat ruhen në sistem.
2. Koha e terminit është zënë.

### Test Case: Contact Us

### Test Case ID: TC002

### Test Case Title: Dërgimi i një mesazhi përmes formës "Contact Us".

Preconditions

1. Përdoruesi duhet të ketë qasje në ëeb aplikacion.
2. Faqja "Contact Us" duhet të jetë e qasshme.
3. Të dhënat e formës duhet të plotësohen në mënyrë korrekte.

Hapat e Testimit

1. Hap faqen "Contact Us":
   * Action: Kliko në butonin "Contact Us”.
   * Rezultatet e Pritura: Faqja "Contact Us" hapet me sukses dhe tregon formën për të dërguar një mesazh.
2. Plotëso të dhënat e formës "Contact Us":
   * Action: Mbush të dhënat e kërkuara në formë:
     + Emri: Shkruan emrin e përdoruesit.
     + Mbiemri: Shkruan mbiemrin e përdoruesit
     + Email-a: Shkruan adresën e email-it të përdoruesit.
     + Numri i Telefonit:Vendos numrin e telefonit të përdoruesit.
     + Mesazhi: Shkruan mesazhin që dëshiron të dërgojë.
   * Rezultatet e Pritura: Të dhënat vendosen saktë dhe forma është gati për dërgim.
3. Dërgo mesazhin:
   * Action: Kliko në butonin "Submit" ose "Send".
   * Rezultatet e Pritura: Një mesazh konfirmimi shfaqet, që tregon se mesazhi është dërguar me sukses dhe pastrohet forma për dërgimin e mesazheve tjera

Postconditions

1. Mesazhi ruhet në sistem për t’u procesuar nga administrata.

*Punuar nga Drilon Zogaj më datë 30.12.2024*