****

Capstone Project Phase B

24-2-D-31

**Booking An Appointment**

**With Advance Payment**

**-NexTor-**

**Members Names:**

**♕ Hanan Madi – 314748211**

**♕ Rawan Sadeq – 315183723**

**Advisor’s name:**

**♕ Zakharia Frenkel**

**Git Repository:**

[**https://github.com/RawanSadeq5/Final-Project.git**](https://github.com/RawanSadeq5/Final-Project.git)

**NexTor Website:**

[**https://nex-tor.onrender.com**](https://nex-tor.onrender.com)

**Table of contents**

Abstract………………………………………………………………………………3

Introduction…………………………………………………………………………..4

General Description…………………………………………………………………..5

Activity Diagram……………………………………………………………………..6

Flow Char…………………………………………………………………………….9

Development Process Description…………………………………………………...10

Tools Used and Client Interface During Development…………………………..….12

Challenges Faced and Solutions……………………………………………………..14

Results and Insights…………………………………………………………..……..16

Lessons Learned…………………………………………………………..…………19

Did We Meet the Project Benchmarks………………………………..……………..21

User Guide…………………………………………………………..………………23

Maintenance Guide…………………………………………………………..………42

**Abstract**

Service based businesses face significant challenges in managing appointments, including high rates of no-shows, last-minute cancellations, and cash flow issues due to delayed payments.

This system effectively streamlines appointment booking, minimizes no-shows, and ensures timely payment collection, providing businesses with a reliable solution to optimize operations and improve cash flow.

The platform incorporates a robust set of features to ensure continuous and reliable scheduling, including automated reminders to reduce missed appointments, dynamic waiting lists to fill cancellations in real time, and flexible payment options that secure commitments in advance. With 24/7 self-service appointment management, customers can conveniently book, reschedule, or cancel appointments, ensuring a smooth and uninterrupted scheduling experience. These capabilities empower business owners to maintain a steady flow of appointments while enhancing customer satisfaction.

Developed using a modern tech stack, including Html, CSS and JavaScript for the front end, Node.js for the back end, MongoDB for data management, the platform ensures scalability, security, and reliability.

**Introduction**

Service based businesses frequently encounter challenges in managing appointments, such as high no-show rates, last-minute cancellations, and inconsistent cash flow caused by delayed payments. These issues disrupt operational efficiency, compromise revenue stability, and strain customer relationships. Recognizing these pain points, we developed NexTor, a cutting-edge website designed to address these challenges and revolutionize the appointment management process.

The website offers a seamless and comprehensive solution that meets the needs of both business owners and their clients. By introducing a streamlined appointment booking process, the system reduces inefficiencies and enhances customer engagement while ensuring timely payment collection. Its prepayment mechanism reduces appointment cancellations, fostering consistent revenue and greater accountability among clients. Additionally, automated reminders and flexible payment options are integrated to improve customer adherence and convenience.

One of the website’s standout features is its ability to allow clients to transfer appointments to others, eliminating the risk of revenue loss while maximizing appointment utilization. Furthermore, it incorporates an innovative referral program, “a-friend-brings-a-friend”, which encourages client referrals and supports business growth.

Through NexTor, service-based businesses can overcome long-standing challenges in appointment management. This website not only optimizes operational efficiency but also enhances the overall customer experience, setting a new standard for the industry.

**General Description**

Our project focuses on developing a Booking an Appointment with Advance Payment System, designed to streamline appointment scheduling for service-based businesses. The website addresses common challenges such as no-shows, last-minute cancellations, and delayed payments by integrating features that ensure both business efficiency and customer satisfaction.

**Key Objectives:**

* **Streamlined Appointment Booking:** The system simplifies the booking process, enabling customers to schedule services quickly and easily.
* **Advance Payment Integration:** Prepayment options reduce cancellations and ensure consistent revenue for businesses.
* **Enhanced Customer Experience:** Features like automated reminders, flexible rescheduling, and self-service management offer convenience and transparency.
* **Business Optimization:** Tools for managing customer interactions, payments, and scheduling help businesses improve service delivery and operational efficiency.

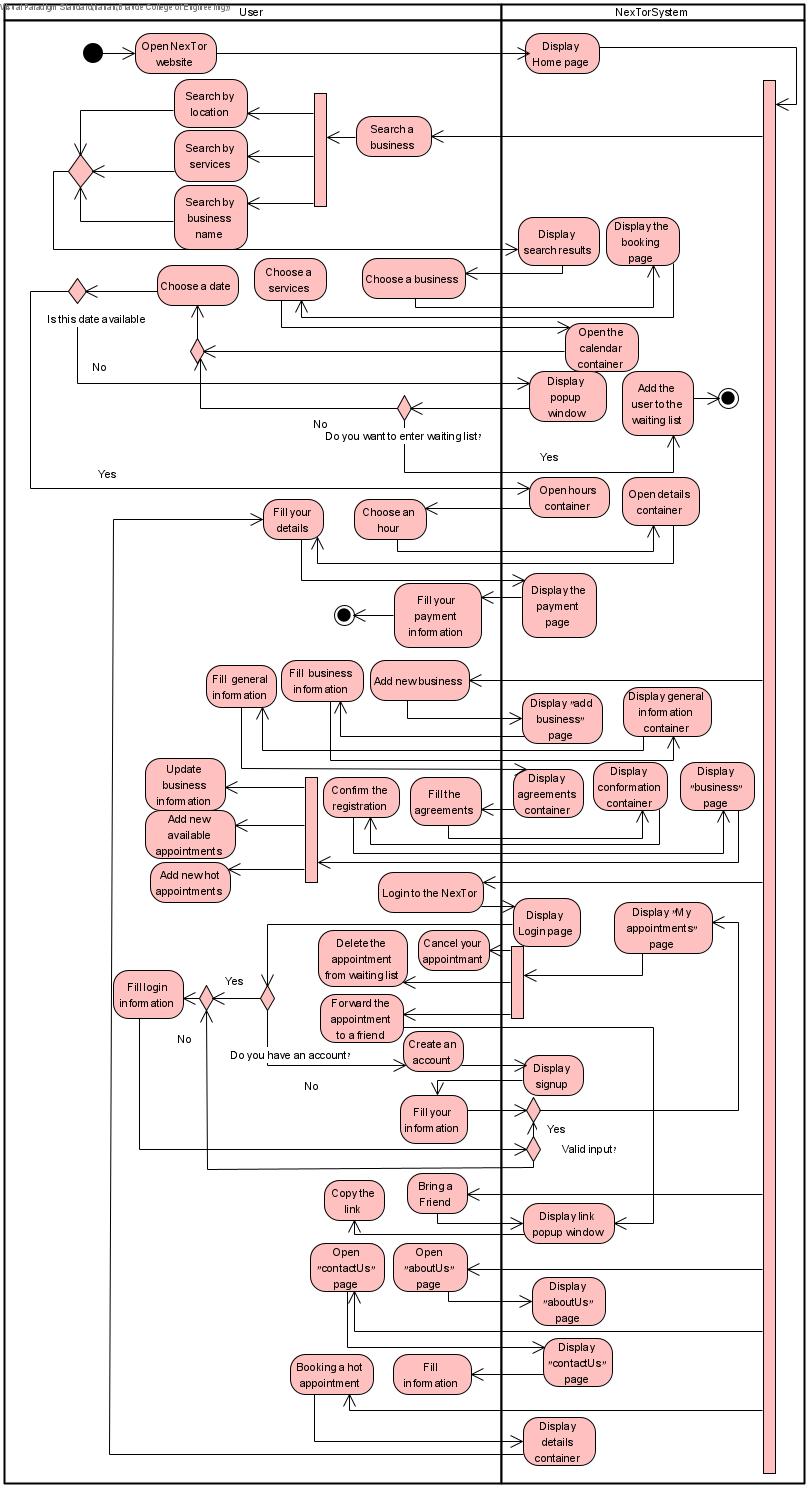
**System Architecture:**

Built on a modern, scalable framework, the website combines a user-friendly interface, secure payment gateways, and robust database management. Key integrations include online payments and calendar synchronization to support seamless operation.

**Target Audience:**

This system is ideal for service-based businesses, such as salons, clinics, consulting firms, and other providers who rely on appointments. It caters to both business owners seeking to optimize operations and customers looking for a reliable and convenient booking experience.

**Activity Diagram**



**Explanation:**

**1. Start and Homepage Entry**

The process begins when the user opens the NexTor website. From the homepage, the user can search for a business by location, services, or business name.

System Action: The NexTor system displays the homepage with search options.

**2. Business Search and Selection**

The user searches for businesses by entering a location, service, or business name.

System Action: The system processes the search query and displays a list of matching results.

The user selects a business from the results.

System Action: The system redirects the user to the booking page for the selected business.

**3. Service and Date Selection**

On the booking page, the user chooses a service and selects a date for the appointment.

System Action: The system checks the availability of the chosen date.

If the date is available, the user can proceed to select a specific time slot.

If the date is unavailable, the system prompts the user to join the waiting list.

**4. Waiting List Option (If Applicable)**

If the user chooses to join the waiting list, they fill out their details, and the system adds them to the waiting list.

System Action: A confirmation is displayed, and the process ends here for the waiting list users.

**5. Time Selection and User Details**

If the date is available, the user selects a specific time slot.

The user provides their personal details (e.g., name, contact information) for the appointment.

System Action: The system processes the details and prepares the payment page.

**6. Payment Information and Validation**

The user is prompted to enter payment details to confirm the appointment.

System Action: The system validates the payment information.

**7. Appointment Confirmation**

Upon successful payment, the system confirms the booking.

**8. User Account Management**

If the user is not logged in, they are prompted to enter their login information before proceeding.

System Action: The system validates the login credentials.

If the credentials are invalid, the system displays an error and prompts the user to try again.

If valid, the user gains access to the booking system and can proceed.

**9. Business Registration and Management**

Business owners can register their business by filling out general and business information.

System Action: The system displays pages for adding and managing business profiles, appointments, and hot appointment slots.

After completing the registration, the business profile is added to the system.

**10. Managing Appointments**

Logged-in users can manage their appointments by canceling, deleting, or forwarding them to friends.

System Action: The system updates the appointment details accordingly and displays updated information.

**11. Additional Features**

Users can access additional features like:

Copying a link to share booking details.

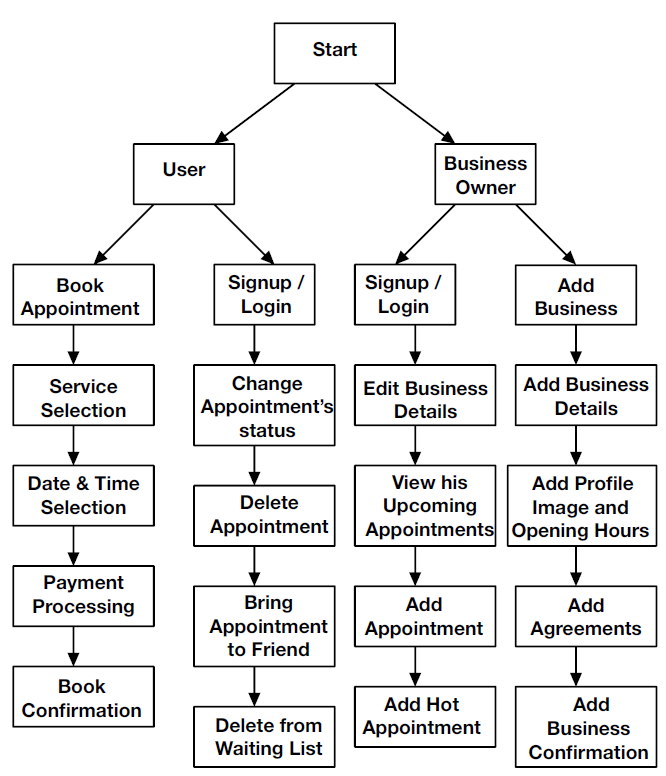
Viewing the "About Us" and "Contact Us" pages for more information about the platform.

**12. End of Process**

The process concludes with the confirmation of an appointment, waiting list addition, or business registration.

The user exits the system after completing their task.

**Flow Chart**

****

This flowchart represents the processes for NexTor website, with distinct paths for users and business owners. Users can book appointments by selecting a service, choosing a date and time, completing payment processing, and receiving a confirmation. They can also manage appointments by changing their status, deleting them, bringing them to friends, or removing them from a waiting list. Business owners, on the other hand, can sign up or log in to manage their business details, such as editing information, adding appointments (including hot appointments), and viewing upcoming schedules. Additionally, they can set up business profiles with images, operating hours, and agreements, finalizing their setup with business confirmation. The chart clearly outlines parallel yet complementary functionalities for both user roles to streamline appointment and business management.

**Development Process Description**

**1.Project Planning and Requirements Gathering**

* **Objective:** Develop a platform for booking appointments with advance payment to enhance operational efficiency, reduce no-shows, and streamline cash flow for service-based businesses.
* **Key Features:**
  + User registration and profile management.
  + Appointment booking with advance payment options.
  + Flexible appointment management, including transfers and cancellations.
  + "A-friend-brings-a-friend" referral system.
  + Waiting list management.

**2.Technology Stack Selection**

* Frontend: Html, CSS and JavaScript.
* Backend: Node.js, Express.js.
* Database: MongoDB.

**3.Database Design**

* **Collections:**
  + User: Stores user details, roles, and authentication data.
  + Businesses: Stores businesses details.
  + Appointment: Tracks appointment details, status, and payments.
  + Waiting List: Manages users waiting for earlier appointments.

**4.Initial Setup**

* **Backend:**
  + Initialized Node.js project with Express, Mongoose, and other necessary modules.
  + Configured database connection and environment variables for sensitive data.
  + Designed modular folder structure for models, controllers, and routes.
* **Frontend:**
  + Developed static and dynamic pages using HTML, CSS, and JavaScript.
  + Integrated APIs for seamless communication with the backend.

**5.Feature Implementation**

* **User Management:**
  + Implemented secure registration and login features with bcrypt password hashing.
  + Developed profile management functionality for business owners and customers.
* **Appointment Booking and Payment:**
  + Created a streamlined booking flow with real-time availability checks.
  + Enabled secure advance payment processing.
* **Advanced Features:**
  + Appointment transfer and cancellation mechanisms with predefined policies.
  + Referral program to encourage website adoption and growth.

**6.Testing and Quality Assurance**

* Conducted extensive unit, integration, and user acceptance testing.
* Ensured cross-browser compatibility and responsiveness across devices.

**Tools Used and Client Interface During Development**

**Frontend (HTML, CSS, and JavaScript-based):**

* **HTML and CSS**:
  + Built the structure and styled the user interface using HTML and CSS.
* **JavaScript**:
  + Developed reusable scripts for common functionalities to improve maintainability.
* **Styling Frameworks**:
  + Applied CSS frameworks like Bootstrap for faster and consistent styling.

**Backend (Node.js-based):**

* **Express.js**:
  + Used Express.js to create a lightweight and efficient backend for handling HTTP requests and serving RESTful APIs.
* **Routing**:
  + Defined routes in the Express app to handle user actions like booking, payments, and notifications.
  + Organized routes into modules for better code maintainability.
* **Authentication and Authorization**:
  + Implemented secure authentication using bcrypt for password hashing.
  + Managed session-based authentication for user login and access control.

**Database:**

* **MongoDB**:
  + Chosen as the database for its flexibility in handling JSON-like documents.
  + Utilized Mongoose for schema design, validations, and streamlined data interactions.

**Deployment:**

1. **Backend (Render):**
   * Deployed the backend using Render, ensuring seamless integration with the Git repository.
   * Automatically built and deployed the latest backend version upon each commit, maintaining an up-to-date server.
2. **Frontend (Render):**

* Deployed the frontend on Render for hosting static content and handling client-side interactions.
* Configured automatic builds and deployments linked to the Git repository.
* Ensured secure access through custom domains and SSL certificates.

**Challenges Faced and Solutions**

1. **Choosing the Right Frontend Framework**
   * **Challenge**: Initially, React was considered for the frontend. However, due to project constraints and the team's familiarity, HTML, CSS, and JavaScript were chosen instead.
   * **Solution**: By leveraging JavaScript for dynamic features and CSS for styling, we efficiently implemented the frontend while avoiding the learning curve of a new framework. This allowed focus on delivering the required functionalities without compromising performance.
2. **User Authentication and Security**
   * **Challenge**: Implementing secure user authentication to protect sensitive information like payment data while ensuring ease of use.
   * **Solution**: Used bcrypt for password hashing and implemented secure session management using express session. Strict input validation was performed to prevent unauthorized access and ensure the security of user data.
3. **Handling Data Validation**
   * **Challenge**: Ensuring the integrity and accuracy of user inputs, such as booking details and payments, was critical to avoid errors and malicious data.
   * **Solution**: Server-side validation was implemented using validation libraries and middleware in Node.js. On the frontend, JavaScript was used for real-time input validation, providing immediate feedback to users.
4. **Integration of Frontend and Backend**
   * **Challenge**: Coordinating data exchange between the HTML/JavaScript frontend and the Node.js/MongoDB backend while handling asynchronous operations.
   * **Solution**: Defined clear API endpoints in the backend for frontend communication.
5. **Handling Asynchronous Operations**
   * **Challenge**: Managing asynchronous tasks like database operations and external API calls to prevent race conditions and ensure proper data handling.
   * **Solution**: Employed async/await syntax for cleaner, more maintainable code. Proper error handling mechanisms were used to catch and resolve issues during asynchronous operations.
6. **Optimizing Performance**
   * **Challenge**: As user data and transactions increased, ensuring the platform remained fast and responsive was critical.
   * **Solution**:
     + Indexed MongoDB collections for faster queries.
     + Used pagination for data-intensive features like appointment history.
     + Minimized frontend asset size by optimizing CSS and JavaScript.

**Results and Insights**

**Project Goals Achievement:**

The primary goal of the project was to develop a robust web platform for booking appointments with advance payment, addressing challenges like no-shows, last-minute cancellations, and cash flow issues. Through structured planning and the use of appropriate technologies, these objectives were successfully met. Here's a breakdown of the achievements:

1. **User Authentication and Security:**
   * **Achievement**: Implemented secure user authentication using bcrypt for password hashing and session-based authentication for user login.
   * **Reasoning**: Ensuring secure access to user accounts was critical. The system was designed to handle sensitive data securely while providing users with a seamless authentication process.
2. **Appointment Booking and Management:**
   * **Achievement**: Developed a dynamic booking system that allowed users to schedule appointments, manage bookings, and make advance payments. Features like waiting lists and appointment transfers.
   * **Reasoning**: Addressing business challenges like no-shows and cancellations required a flexible yet reliable booking and payment system. The waiting list and transfer features provided users with added convenience, while advance payments ensured business owners received steady revenue.
3. **Seamless User Experience:**
   * **Achievement**: Designed a responsive and intuitive user interface using HTML, CSS, and JavaScript. Booking flows, payment processes, and profile management were streamlined for ease of use.
   * **Reasoning**: A focus on user experience ensured higher engagement and satisfaction, making the platform accessible and user-friendly for both customers and business owners.

**Coping with Challenges:**

The project encountered various challenges, which were addressed through strategic planning and innovative solutions:

1. **Technology Selection:**
   * **Challenge**: Choosing the right frontend technology while balancing performance and development efficiency.
   * **Coping Strategy**: Opted for HTML, CSS, and JavaScript, which allowed rapid development without the complexity of additional frameworks like React. This choice aligned with the team's expertise and ensured timely delivery.
2. **Security and Authentication:**
   * **Challenge**: Implementing a secure and user-friendly authentication system.
   * **Coping Strategy**: Used bcrypt for secure password hashing and express session for session management. Sensitive data was stored securely using environment variables to prevent unauthorized access.
3. **Performance Optimization:**
   * **Challenge**: Ensuring the application remained responsive with increasing user data and transactions.
   * **Coping Strategy**: Implemented MongoDB indexing for faster queries, used pagination for data-intensive pages, and optimized JavaScript code to reduce load times.
4. **Error Handling and User Feedback:**
   * **Challenge**: Providing meaningful and actionable feedback to users in case of errors.
   * **Coping Strategy**: Developed robust error handling on both frontend and backend. Users received clear messages and guidance, ensuring smooth recovery from errors.

**Decision-Making Considerations:**

Several key factors influenced decision-making throughout the project:

1. **Scalability and Flexibility:**
   * Technologies and designs were chosen to accommodate future growth, such as MongoDB for its schema flexibility and modular backend architecture for easy feature expansion.
2. **Security:**
   * Ensuring data security was a top priority. Decisions on authentication, payment processing, and data storage were guided by best practices and industry standards.
3. **User Experience:**
   * Creating an intuitive, visually appealing, and responsive platform was central to the project. This focus drove decisions design and frontend implementation.
4. **Maintainability:**
   * A clean and organized codebase was maintained, with modular structures and reusable components. This approach ensured long-term maintainability and ease of future updates.

**Lessons Learned**

**1. Flexibility in Technology Choices**

**Lesson:** The selection of technology tools and frameworks for NexTor highlighted the importance of adaptability. Choosing technologies that suited the requirements of a booking system with advanced payment capabilities was pivotal.

**Takeaway:** Always evaluate the project's evolving needs and adjust technology choices as necessary to optimize functionality and scalability.

**2.Importance of Thorough Planning**

**Lesson:** Comprehensive planning was crucial for understanding user needs, defining workflows for appointment booking and payment integration, and designing the overall system.

**Takeaway:** Invest time in planning and requirement gathering to anticipate challenges and align system components effectively. This reduces rework and ensures smooth development.

**3.Security Best Practices**

**Lesson:** Implementing secure payment gateways and user authentication systems emphasized the importance of robust security measures to protect sensitive data.

**Takeaway:** Security is non-negotiable. Prioritize secure coding practices, such as encrypting payment details and protecting personal data, to maintain user trust and compliance with regulations.

**4.User-Centric Design**

**Lesson:** A focus on creating an intuitive booking interface with clear payment options was instrumental in enhancing user satisfaction.

**Takeaway:** Always prioritize the end-user's perspective. A user-friendly and accessible design drives engagement and improves the overall experience.

**5.Effective Error Handling**

**Lesson:** Building comprehensive error-handling mechanisms ensured users received clear and actionable feedback, reducing frustration during booking and payment processes.

**Takeaway:** Clear and user-focused error handling simplifies debugging and enhances user trust in the system.

**6.Collaboration and Communication**

**Lesson:** Close collaboration between developers, designers, and stakeholders was essential for aligning project goals and tackling challenges like payment integrations and scheduling features.

**Takeaway:** Foster open communication and teamwork to ensure the project stays aligned with its objectives and meets all stakeholder expectations.

**7.Iterative Development and Testing**

**Lesson:** Adopting an iterative approach enabled continuous testing of booking flows, payment integrations, and notification features, identifying issues early in the process.

**Takeaway:** Regular testing and refinement improve reliability and result in a robust final product.

**8.Handling Complexity with Modular Design**

**Lesson:** Breaking the system into modular components such as appointment management, payment processing, and notifications simplified development and facilitated scalability.

**Takeaway:** Use modular design principles to handle system complexity, enabling easier updates and maintenance for future improvements.

These lessons from developing NexTor will guide future projects, helping us streamline processes, enhance collaboration, and deliver exceptional systems tailored to user needs.

**Did We Meet the Project Benchmarks?**

We are confident that we successfully met the benchmarks set for the NexTor project at its inception. Below is a detailed breakdown of our goals and achievements:

**1.Implementation of User Authentication:**

**Benchmark:** Develop a secure and user-friendly authentication system to manage both user accounts and admin access.

**Outcome:** We implemented a secure authentication mechanism using bcrypt for password hashing and session management. The system provides seamless registration and login experiences while safeguarding user data.

**2.Appointment Booking System:**

**Benchmark:** Create a robust and intuitive booking system that allows users to schedule appointments and manage bookings efficiently.

**Outcome:** The booking system includes features like calendar integration and real-time slot availability, delivering a smooth and efficient scheduling experience for users.

**3.Advance Payment Integration:**

**Benchmark:** Ensure the platform securely processes advance payments during appointment bookings.

**Outcome:** We successfully integrated secure payment gateways to handle advance payments.

**4.Database Management:**

**Benchmark:** Implement a database solution capable of handling appointment records, user data, and payment information effectively.

**Outcome:** MongoDB was chosen for its flexibility and scalability, enabling efficient management of complex data structures and ensuring reliable performance.

**5.Error Handling and Validation:**

**Benchmark:** Incorporate robust error handling and input validation to ensure a seamless user experience and system reliability.

**Outcome:** Comprehensive error messages and validation checks were implemented throughout the system. These features enhanced usability and ensured the platform handled unexpected scenarios gracefully.

**6.Performance and Scalability:**

**Benchmark:** Optimize the system for fast response times and prepare it to handle increased user activity as the platform grows.

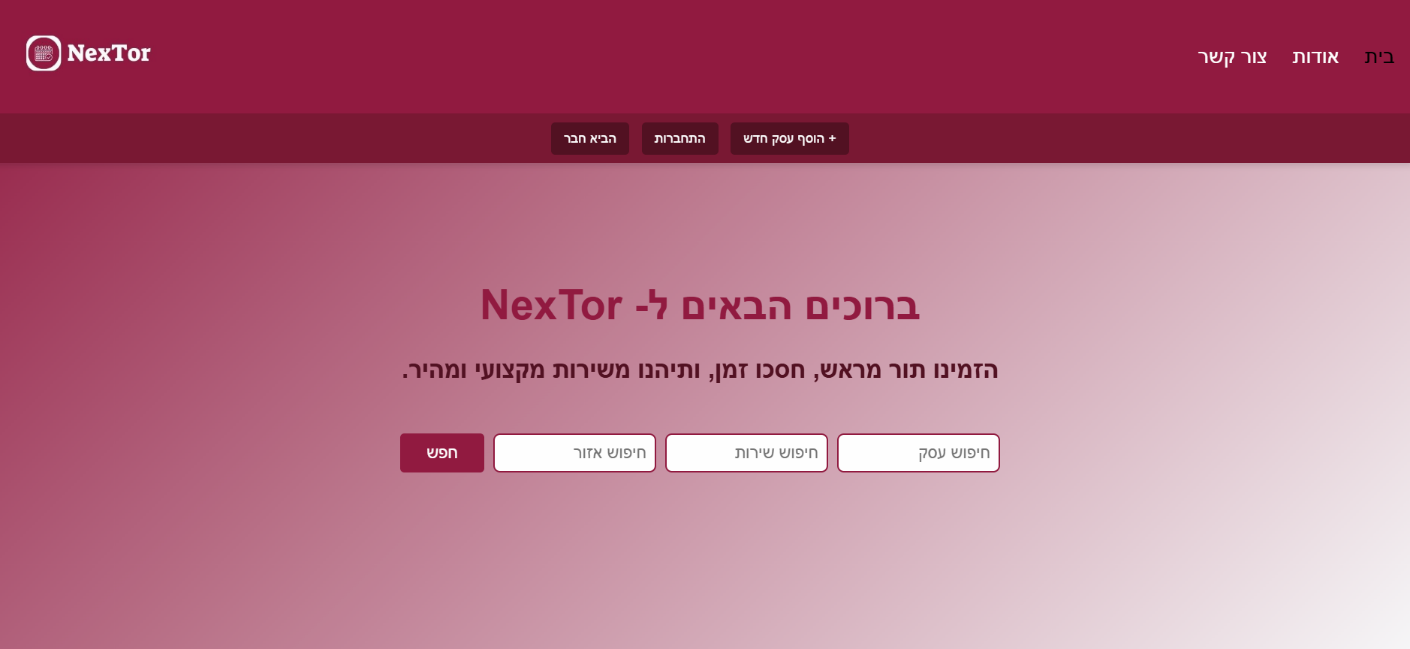
**Outcome:** By optimizing database queries and leveraging modular design principles, the system achieves excellent performance and is prepared for future scalability.

**7.Security and Privacy:**

**Benchmark:** Implement industry-standard security practices to protect user data, payments, and session information.

**Outcome:** Security was prioritized with measures such as encrypted payment processing, secure session management, and adherence to data protection standards, ensuring user trust and compliance.

**User Guide**

**Home screen:**

**עמוד הבית של :NexTor**

**סרגל הניווט העליון -** החלק העליון של עמוד הבית מאפשר למשתמשים לנווט בין אפשרויות ותכונות שונות:

**בית:** מעביר את המשתמשים לעמוד הראשי של האתר.

**אודות:** מספק מידע עלNexTor , המטרות שלו והתכונות שהוא מציע.

**צור קשר**: מוביל לעמוד שבו ניתן ליצור קשר עם צוות NexTor לצורך תמיכה או שאלות.

**הוסף עסק חדש:** תכונה לבעלי עסקים המאפשרת להוסיף את העסק שלהם לאתר.

**התחברות:** מאפשרת למשתמשים קיימים להתחבר ולגשת לחשבון האישי שלהם.

**הבא חבר:** מספק גישה לאפשרות להפנות חברים.

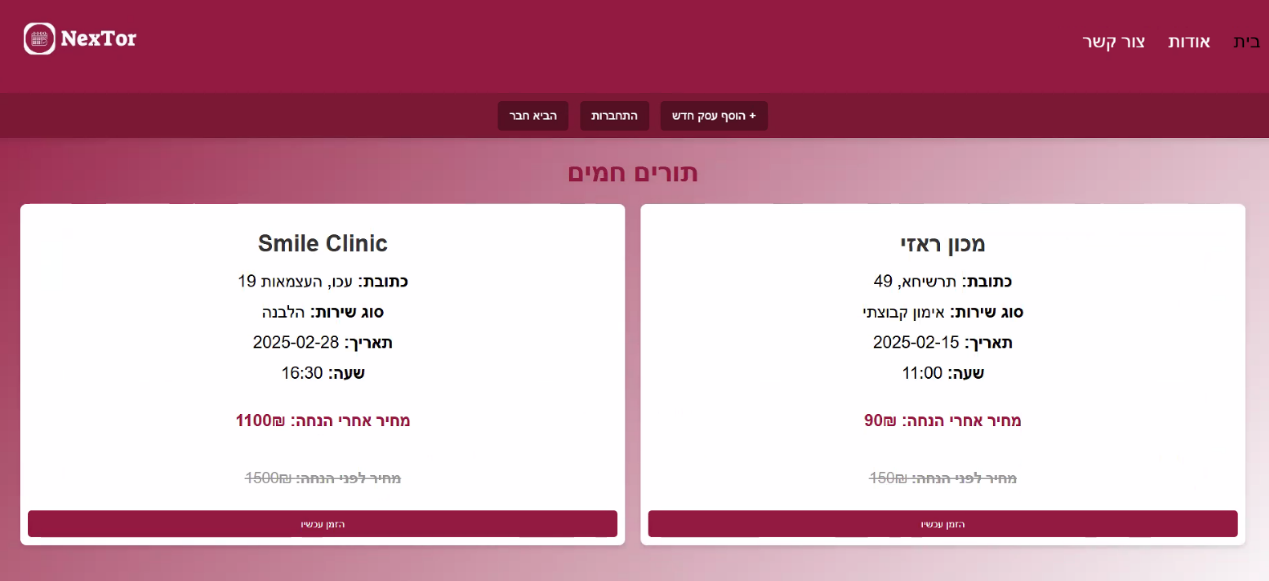
**החלק המרכזי -** החלק המרכזי של עמוד הבית נועד להנחות את המשתמשים בחיפוש עסקים ושירותים:

**חיפוש עסק:** מאפשר למשתמשים לחפש עסק מסוים לפי שם.

**חיפוש שירות:** מאפשר למשתמשים לחפש עסקים לפי שירותים ספציפיים שהם מציעים.

**חיפוש אזור:** מאפשר למשתמשים למצוא עסקים לפי מיקום גיאוגרפי.

**כפתור חיפוש:** מפעיל את החיפוש בהתבסס על הקריטריונים שנבחרו.

**התורים החמים בעמוד הבית:**

תורים חמים באתר NexTor מציג אפשרויות לתורים מיוחדים זמינים בהנחות. ניתן להשתמש בעמוד זה להזמין תורים במהירות וביעילות.

**פרטי התור המוצגים:**

**שם העסק:** מציין את שם העסק המציע את השירות.

**כתובת:** הכתובת הפיזית של העסק שבו השירות יינתן.

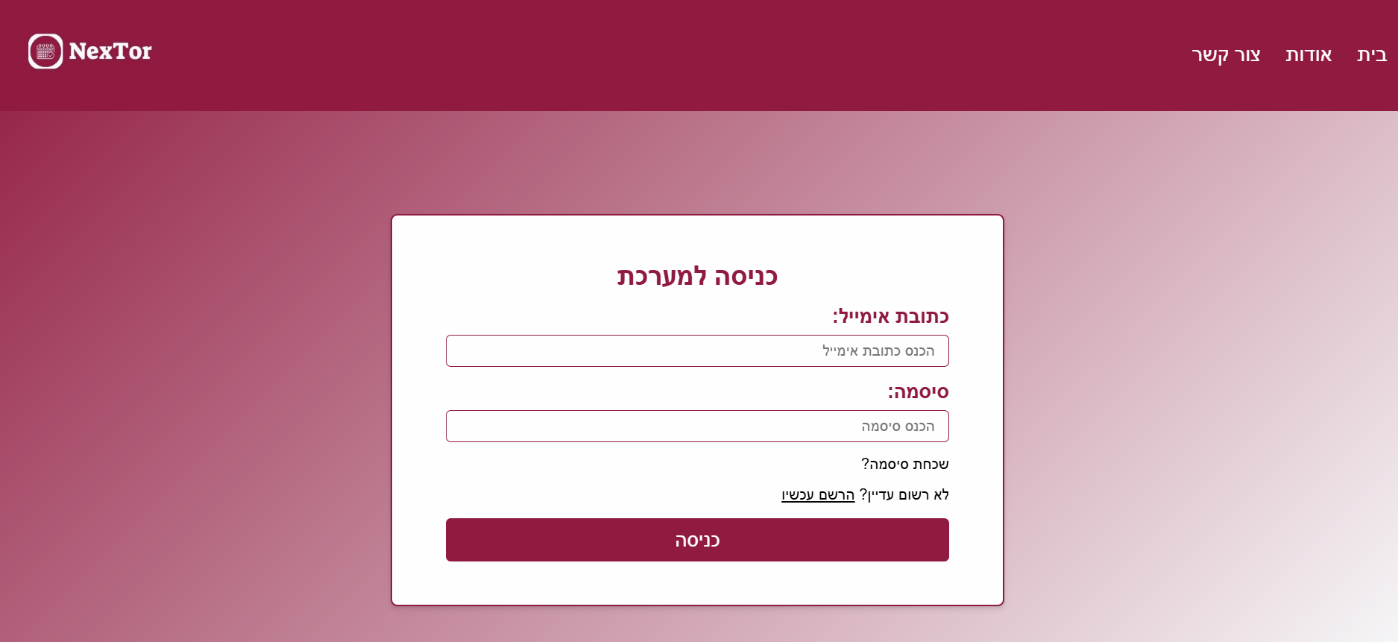
**סוג השירות:** תיאור השירות המוצע.

**תאריך ושעה:** תאריך ושעת ההתחלה של התור הזמין.

**משך השירות:** הזמן המוערך לביצוע השירות.

**מחיר:** המחיר המוזל של השירות. המחיר המקורי מוצג בקו מחוק.

**כפתור "הזמן עכשיו":** בלחיצה על הכפתור, נפתח קונטיינר למלא הפרטים של הלקוח ומועבר לתשלום המקדימה להשלמת ההזמנה של התור הנבחר.

**Login screen:**

**עמוד ההתחברות של :NexTor**

עמוד זה מאפשר למשתמשים להתחבר לחשבון האישי שלהם במערכת .NexTor

**שדות הזנה:**

**כתובת אימייל:** המשתמש מתבקש להזין את כתובת האימייל הרשומה בחשבון.

**סיסמה:** המשתמש מתבקש להזין את הסיסמה האישית שלו.

**כפתור כניסה:** לאחר מילוי כתובת האימייל והסיסמה, המשתמש לוחץ על הכפתור כדי להתחבר לחשבון שלו.

**אפשרויות נוספות:**

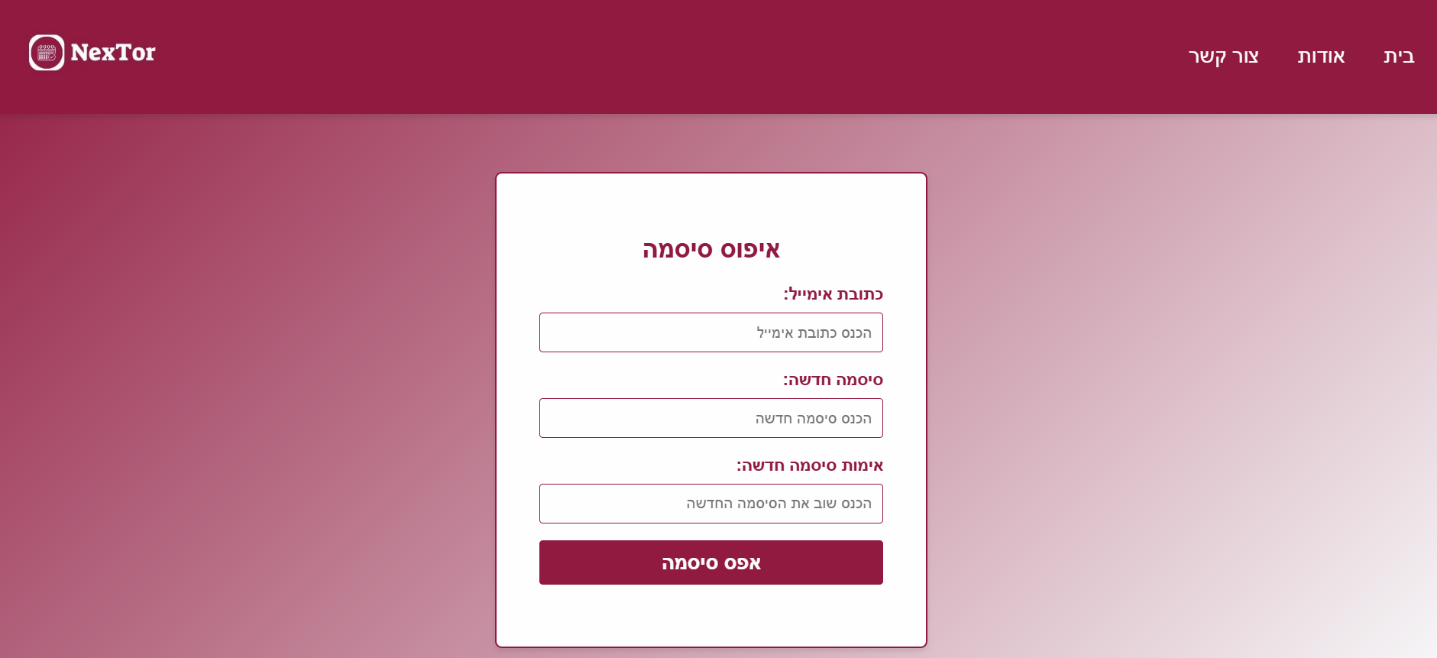
**שכחת סיסמה?** קישור המאפשר למשתמשים לשחזר את הסיסמה במקרה ששכחו אותה.

**לא רשום עדיין? הרשם עכשיו:** קישור שמוביל לעמוד ההרשמה למשתמשים חדשים המעוניינים ליצור חשבון במערכת.

**הערות למשתמש:**

יש לוודא שכתובת האימייל והסיסמה הוזנו בצורה נכונה כדי להימנע מטעויות בהתחברות.

במקרה של בעיות טכניות, ניתן לפנות לעמוד "צור קשר" בסרגל הניווט העליון.

**Forget password screen:**

**עמוד איפוס סיסמה:**

עמוד זה מאפשר למשתמשים לאפס את הסיסמה שלהם במקרה ששכחו או מעוניינים לשנות אותה.

**שדות הזנה:**

**כתובת אימייל:** המשתמש מתבקש להזין את כתובת האימייל הרשומה בחשבון.

**מטרה:** לאמת שהבקשה לאיפוס שייכת למשתמש הנכון.

**סיסמה חדשה:** המשתמש מתבקש להזין סיסמה חדשה עבור החשבון.

**הערה:** יש לבחור סיסמה חזקה כדי לשפר את אבטחת החשבון.

**אימות סיסמה חדשה:** המשתמש נדרש להזין שוב את הסיסמה החדשה כדי לוודא שהוזנה בצורה נכונה.

**כפתור "אפס סיסמה":** לאחר מילוי השדות, המשתמש לוחץ על הכפתור כדי לבצע את תהליך האיפוס.

**תהליך האיפוס:**

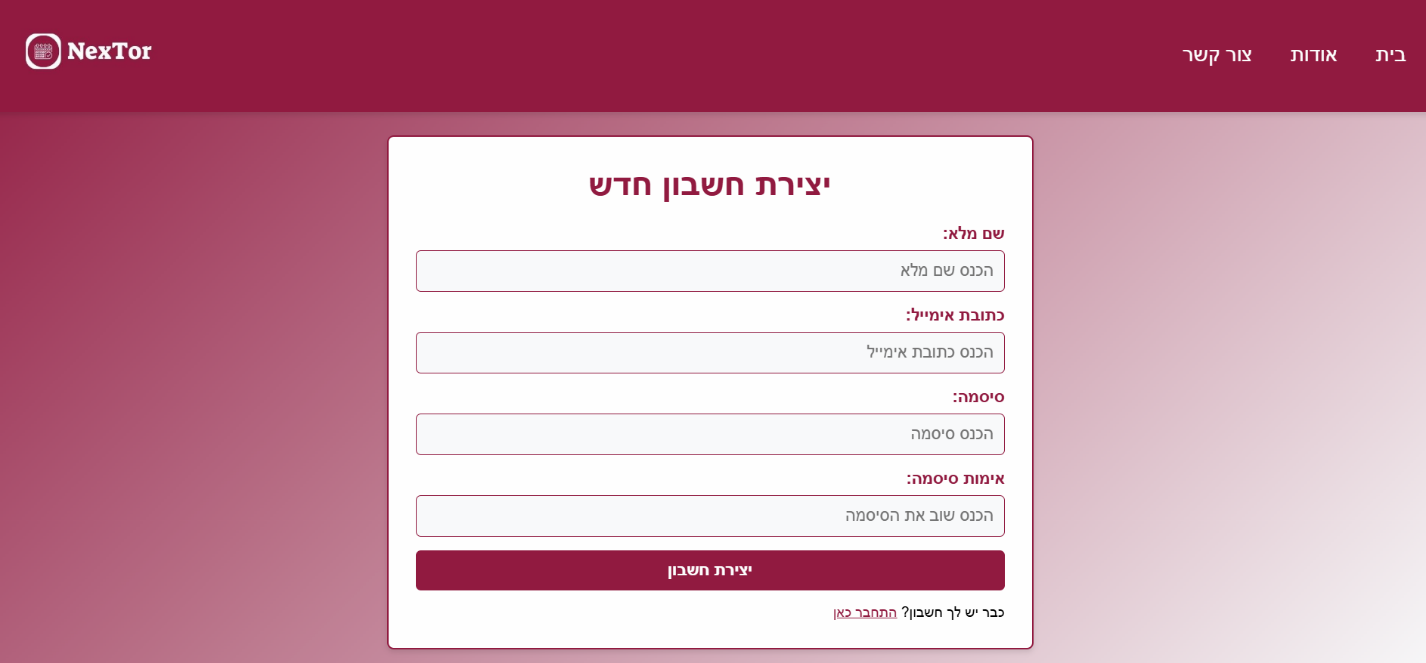
אם כל המידע נכון, המערכת תציג הודעה על הצלחת תהליך האיפוס.

במקרים של טעויות (לדוגמה, כתובת אימייל לא רשומה או סיסמאות שלא תואמות), תוצג הודעת שגיאה עם הסבר.

**הערות למשתמש:**

הקפד למלא את כל השדות בצורה נכונה. בחר סיסמה חדשה חזקה המורכבת מאותיות, מספרים וסימנים.

אם מתעוררות בעיות, ניתן לפנות לעמוד "צור קשר" בסרגל הניווט העליון.

**Signup screen:**

**עמוד יצירת חשבון חדש:**

עמוד זה מאפשר למשתמשים חדשים להירשם וליצור חשבון אישי במערכת NexTor.

**שדות הזנה:**

**שם מלא:** המשתמש מתבקש להזין את שמו המלא.

**כתובת אימייל:** המשתמש מתבקש להזין כתובת אימייל תקפה לצורך הרשמה ואימות חשבון.

**סיסמה:** המשתמש מתבקש לבחור סיסמה אישית לחשבון.

**הערה:** מומלץ לבחור סיסמה חזקה הכוללת אותיות, מספרים וסימנים.

**אימות סיסמה:** המשתמש נדרש להזין שוב את הסיסמה שבחר כדי לוודא שהיא הוזנה בצורה נכונה.

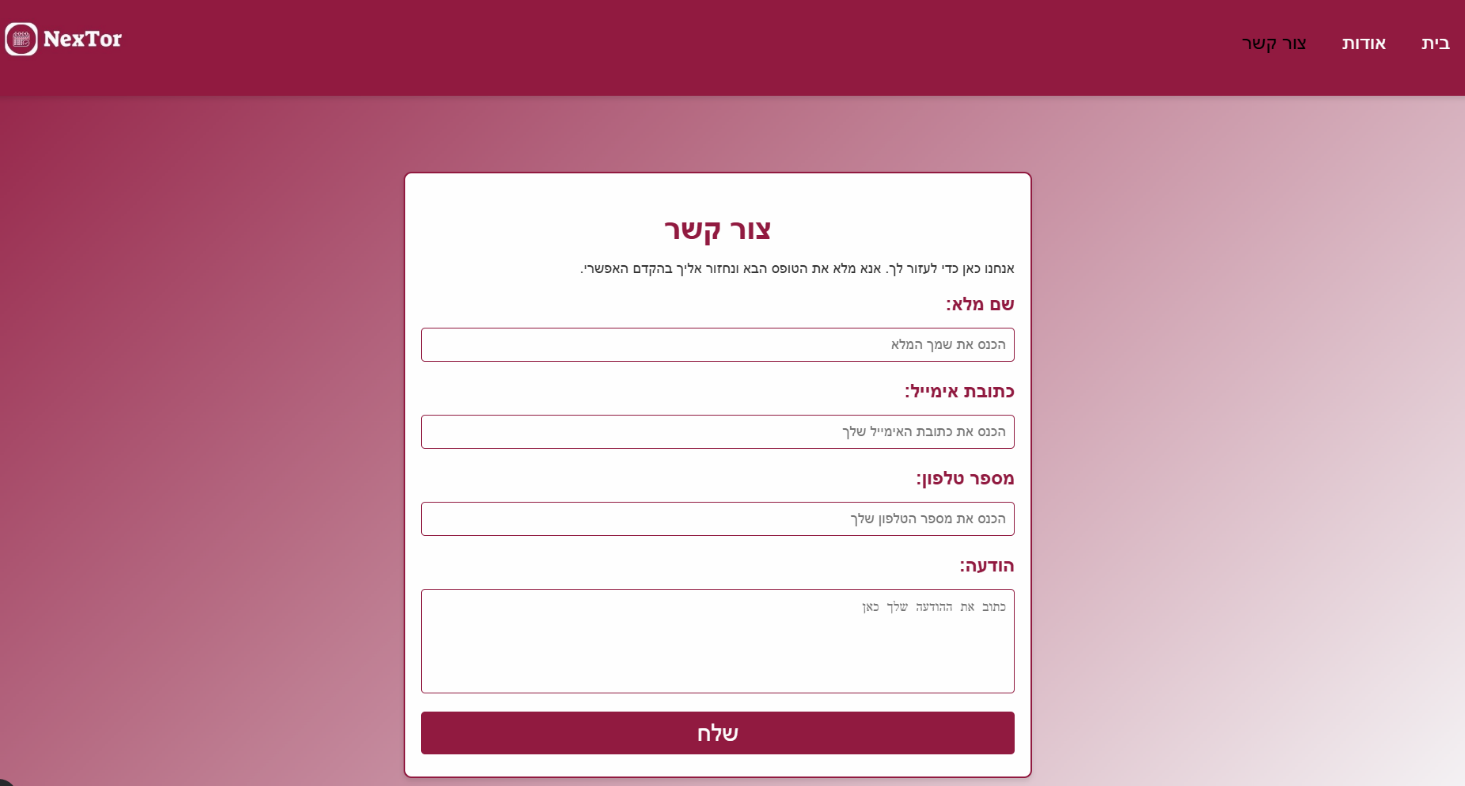
**כפתור "יצירת חשבון":** לאחר מילוי כל השדות, המשתמש לוחץ על הכפתור כדי לסיים את תהליך ההרשמה.

**כבר יש לך חשבון? התחבר כאן:** קישור המוביל לעמוד ההתחברות למשתמשים שכבר רשומים במערכת.

**הערות למשתמש:**

יש למלא את כל השדות בצורה נכונה כדי למנוע טעויות בתהליך ההרשמה.

במקרים של בעיות טכניות, ניתן לפנות לעמוד "צור קשר" שבסרגל הניווט העליון.

**Contact us screen:**

**עמוד צור קשר:**

עמוד זה מאפשר למשתמשים ליצור קשר עם צוות NexTor במקרה של שאלות, בקשות או תמיכה.

**שדות הזנה:**

**שם מלא:** המשתמש מתבקש להזין את שמו המלא.

**כתובת אימייל:** המשתמש מתבקש להזין כתובת אימייל תקפה.

**מספר טלפון:** המשתמש מתבקש להזין את מספר הטלפון שלו.

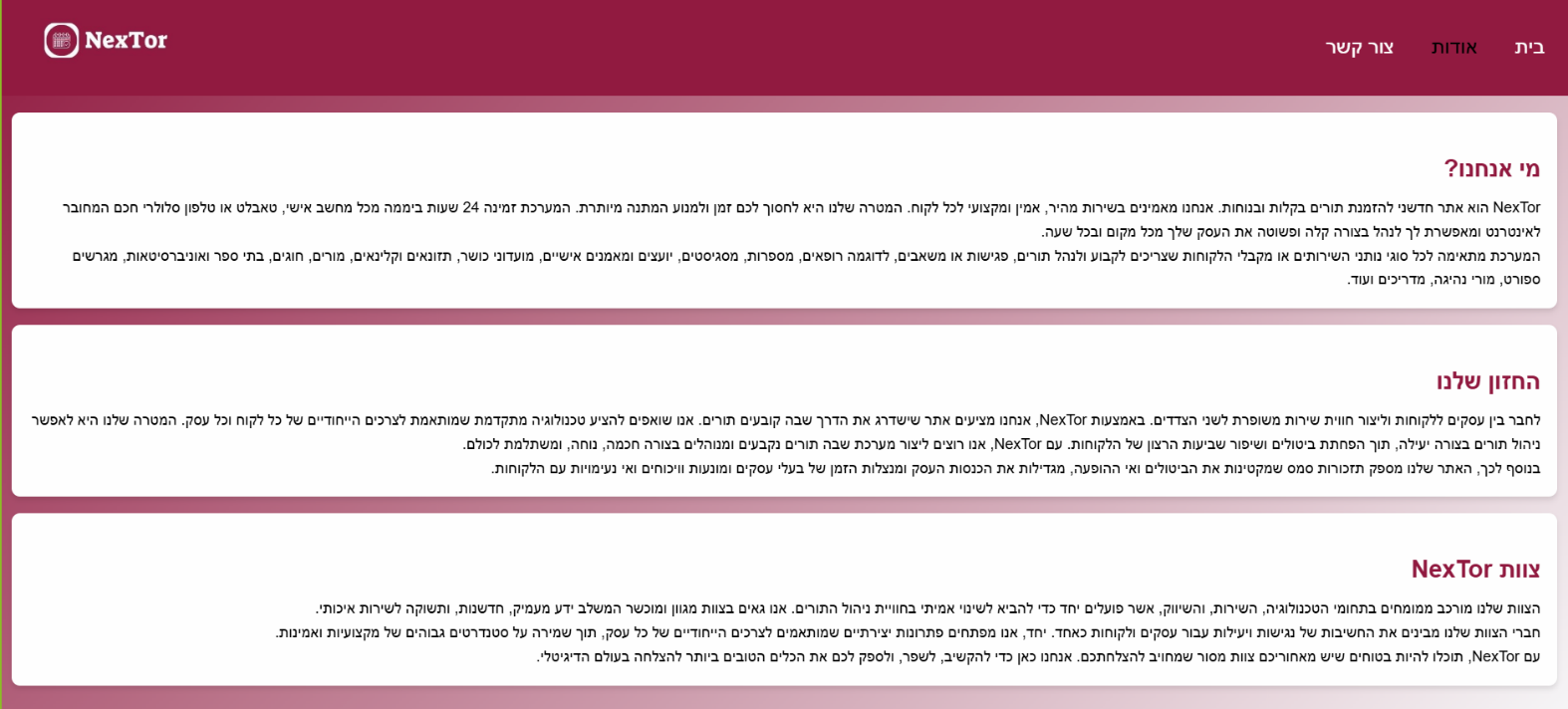
**הודעה:** המשתמש מתבקש לכתוב את תוכן הבקשה או השאלה שלו.

**כפתור "שלח":** לאחר מילוי כל השדות, המשתמש לוחץ על כפתור "שלח" כדי לשלוח את ההודעה לצוות NexTor.

**הערות למשתמש:**

יש למלא את כל השדות בצורה נכונה כדי להבטיח מענה מהיר ויעיל.

במידה ויש צורך בסיוע דחוף, ניתן לפנות גם למוקד התמיכה (פרטים מופיעים בעמוד).

**About us screen:**

**עמוד אודות:**

עמוד זה מספק למשתמשים מידע על NexTor, חזונה, מטרותיה, ועל הצוות העומד מאחוריה.

**מי אנחנו?** NexTor הוא אתר להזמנת תורים בקלות ונוחות, המיועד לכל סוגי השירותים כמו רופאים, מאמנים אישיים, מורים, ואנשי מקצוע נוספים.

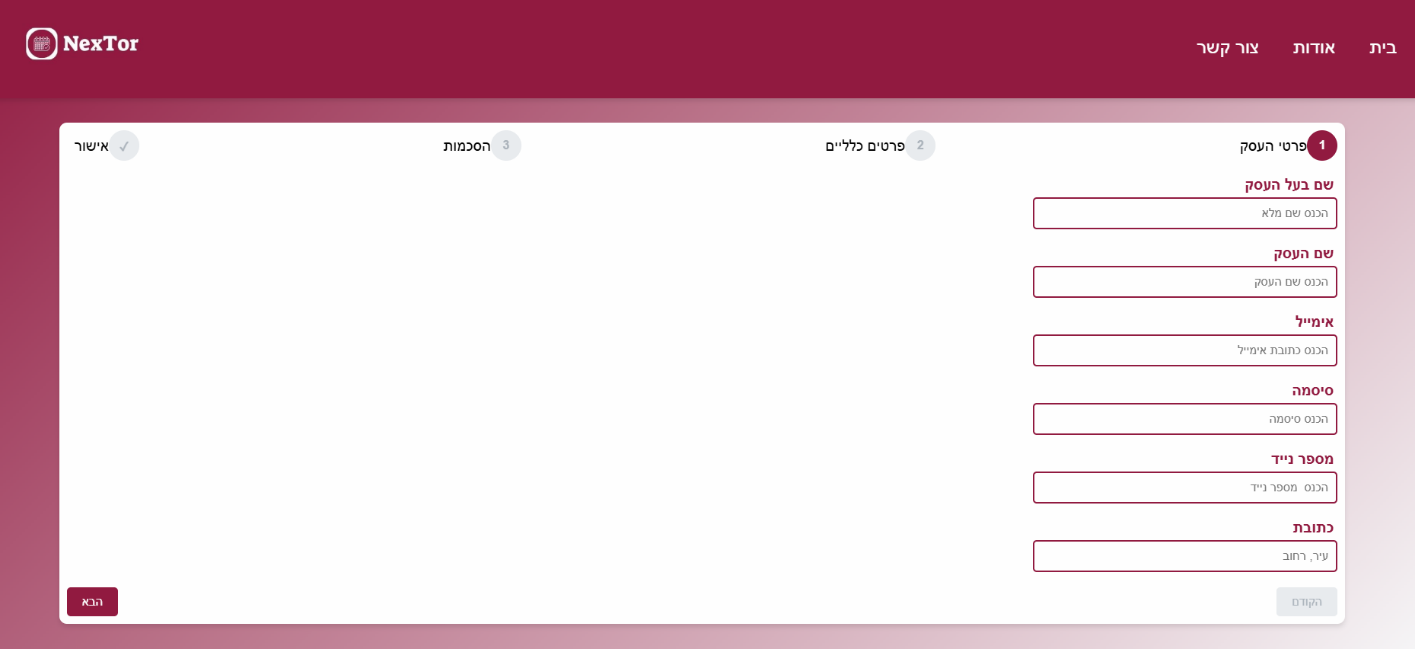
**החזון שלנו:** NexTor שואפת לייצר חווית שירות משופרת לשני הצדדים באמצעות טכנולוגיה מתקדמת.

**צוות NexTor:** הצוות מורכב ממומחים בתחומי הטכנולוגיה, השיווק והשירות, המחויבים לספק את השירות הטוב ביותר למשתמשי NexTor.

**הערות נוספות:**

עמוד "אודות" נועד להציג את הערכים והתפיסה שמנחים את NexTor.

למידע נוסף, ניתן לעיין בעמודים "צור קשר" בסרגל הניווט העליון.

**Add business screen:**

**עמוד הוספת עסק חדש:**

עמוד זה מאפשר לבעלי עסקים לרשום את העסק שלהם באתר NexTor. זהו השלב הראשון בתהליך הרישום הכולל של העסק.

**שדות הזנה:**

**שם בעל העסק:** המשתמש מזין את שמו המלא.

**שם העסק:** המשתמש מזין את שם העסק שיירשם במערכת.

**אימייל:** המשתמש מזין כתובת אימייל תקפה לצורך תקשורת.

**סיסמה:** המשתמש בוחר סיסמה אישית עבור גישה לממשק ניהול העסק.

**מספר נייד:** המשתמש מזין את מספר הטלפון הנייד שלו.

**כתובת:** המשתמש מזין את כתובת העסק (עיר ורחוב).

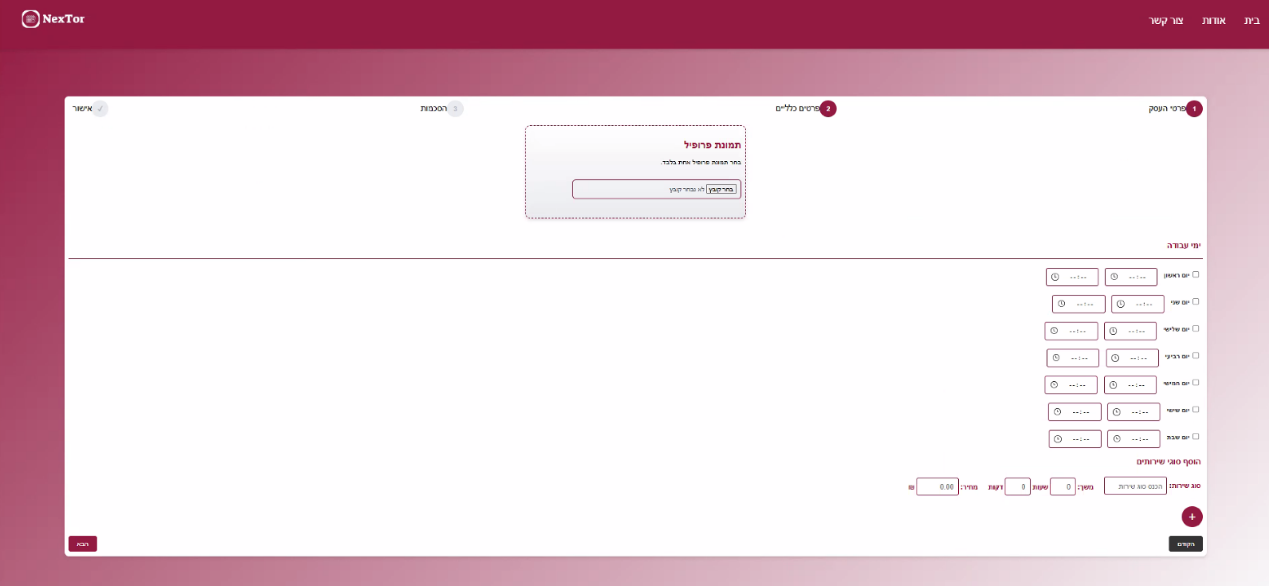
**כפתור "הבא":** לאחר מילוי כל השדות, המשתמש לוחץ על הכפתור "הבא" כדי לעבור לשלב הבא בתהליך הרישום.

**כפתור "הקודם":** מחזיר את המשתמש לשלב הקודם בתהליך הרישום.

**הערות למשתמש:**

יש למלא את כל השדות בצורה נכונה כדי להבטיח רישום מדויק ומהיר.

במידה ויש בעיות טכניות או שאלות, ניתן לפנות לעמוד "צור קשר" דרך סרגל הניווט העליון.

**פרטיים כללים בעמוד הוספת עסק חדש:**

עמוד זה מאפשר לבעלי עסקים להזין מידע כללי וחשוב על העסק שלהם כחלק מתהליך הרישום למערכת NexTor.

**שדות ותכונות בעמוד:**

**תמונת פרופיל:** אפשרות להעלות תמונת פרופיל לעסק.

**כפתור "בחר תמונה":** בוחרים תמונה מהמחשב או המכשיר.

**ימי עבודה:** בחירת הימים שבהם העסק פועל והגדרת שעות הפעילות עבור כל יום עבודה.

**הוספת סוגי שירותים:**

שם השירות, משך השירות ומחיר השירות.

כפתור "+": להוספת שירותים נוספים.

כפתור "-": להסרת שירותים אחרונים.

כפתור "הבא": מעביר את המשתמש לשלב הבא ברישום העסק.

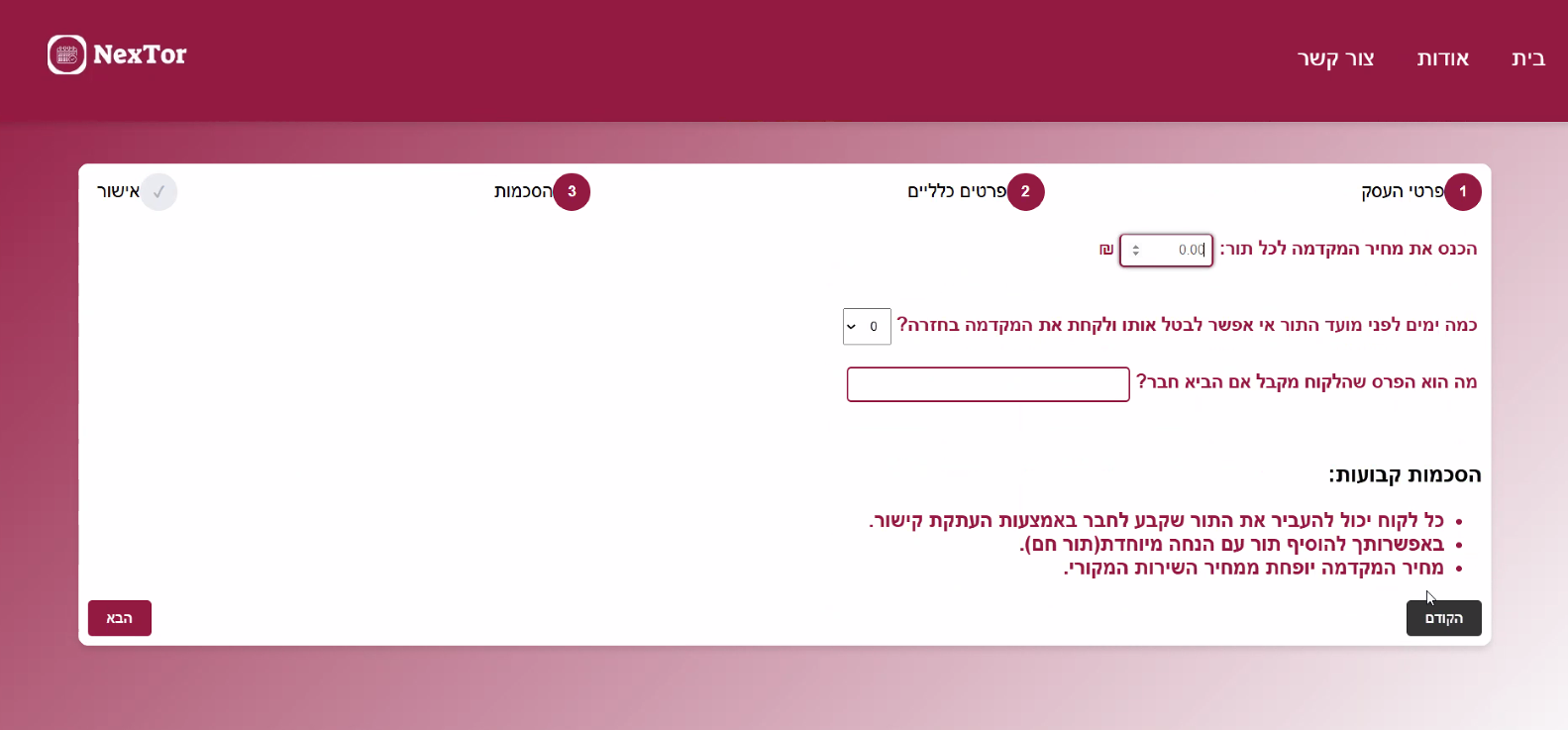
כפתור "הקודם": מחזיר את המשתמש לשלב הקודם אם נדרשת עריכה או עדכון של פרטים.

**הערות למשתמש:**

יש למלא את כל השדות בצורה מדויקת ומלאה כדי להבטיח הצגה מיטבית של העסק במערכת.

במקרה של בעיות טכניות, ניתן לפנות לעמוד "צור קשר" לקבלת סיוע.

מומלץ לוודא שהתמונות המועלות והפרטים שהוזנו מייצגים את העסק בצורה מקצועית.

**הסכמות בעמוד הוספת עסק חדש:**

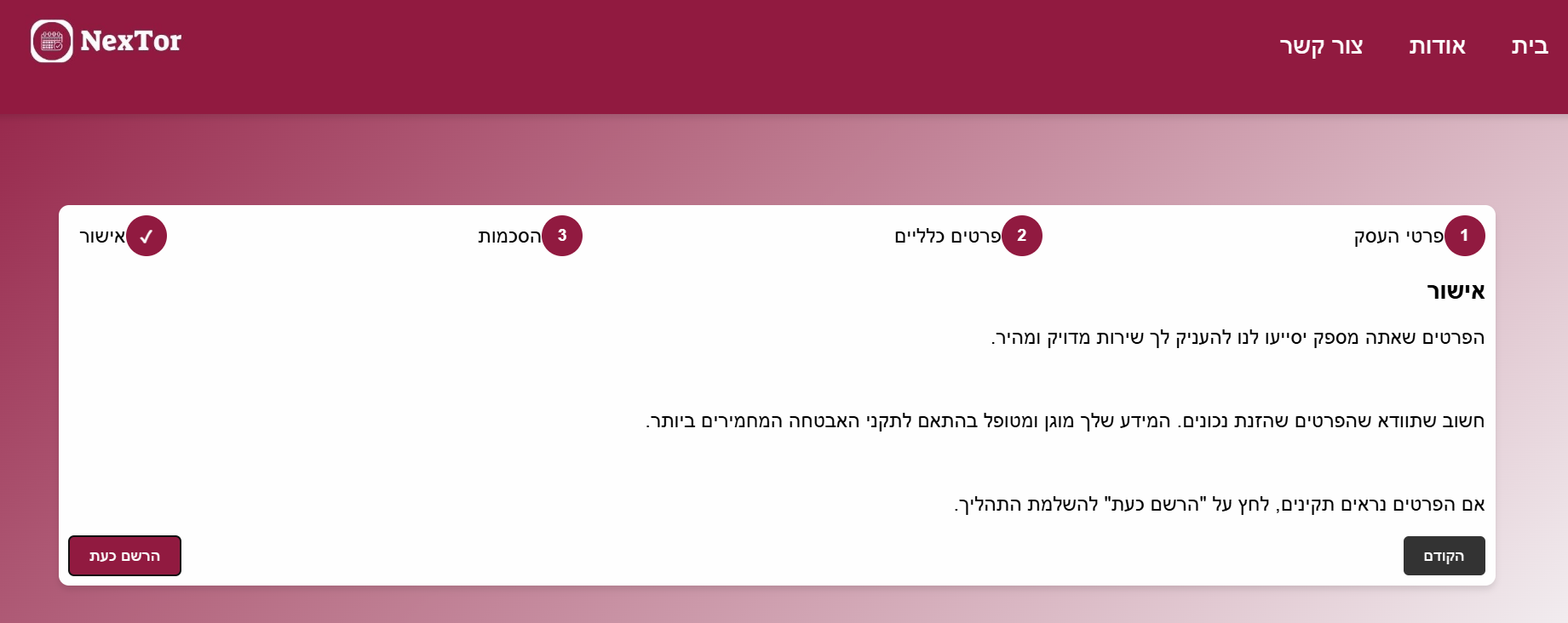
עמוד זה מאפשר לבעלי עסקים לקבוע תנאים והסכמות הקשורות להזמנת תורים במערכת NexTor.

**הערות למשתמש:**

יש לקרוא היטב את כל השדות ולמלאם בהתאם למדיניות העסק.

במקרה של שאלות או בעיות טכניות, ניתן לפנות לעמוד "צור קשר".

מומלץ לחשוב על תנאים הוגנים ונוחים ללקוחות כדי לשפר את חוויית השירות.



**האישור בעמוד הוספת עסק חדש:**

עמוד זה מסכם את פרטי העסק שהוזנו על ידי המשתמש במהלך תהליך הרישום. המידע בעמוד זה מאפשר למשתמש לוודא את נכונות הפרטים לפני השלמת הרישום.

**כפתור "הרשם כעת":** משלים את תהליך הרישום ומוסיף את העסק למערכת NexTor.

**כפתור "הקודם":** מחזיר את המשתמש לעמוד הקודם במידה והוא מעוניין לערוך את הפרטים שהוזנו.

**הערות למשתמש:**

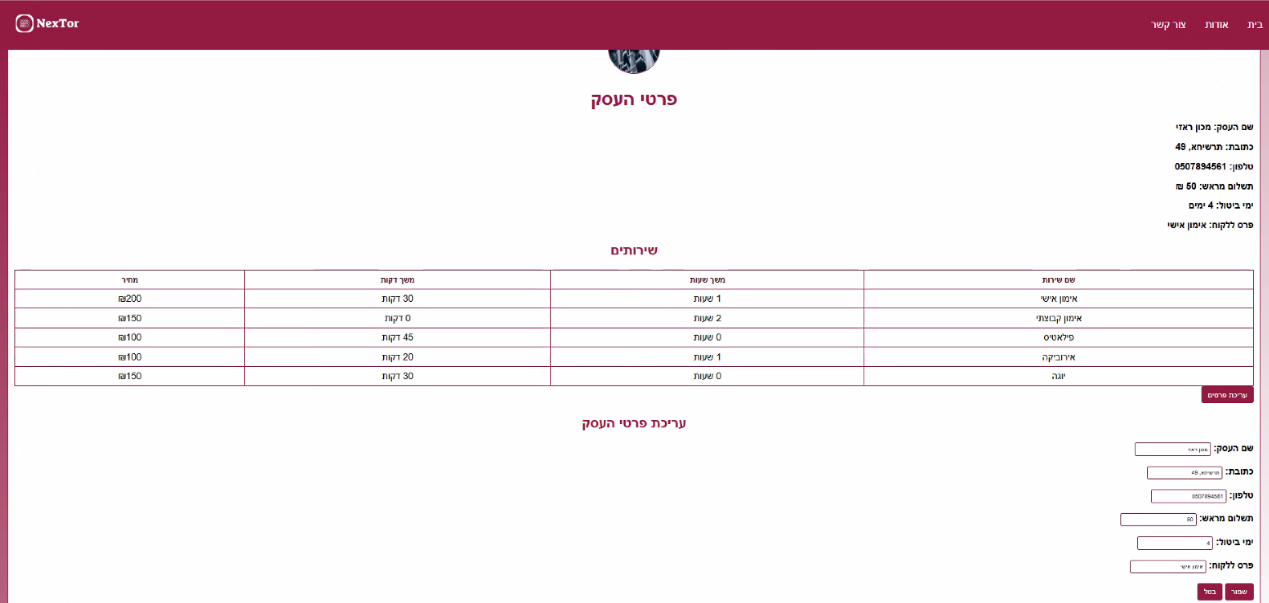
יש לקרוא בקפידה את הפרטים שהוזנו ולוודא את נכונותם לפני לחיצה על "הרשם כעת".

במידה ויש צורך בתיקונים, ניתן לחזור לשלבים הקודמים ולערוך את המידע.

במקרה של שאלות או בעיות, ניתן לפנות לעמוד "צור קשר" בסרגל הניווט העליון.

כאשר לחצת על "הרשם כעת" נפתח לך המסך האישי של העסק שלך.

**Business screen:**

**עמוד בעל העסק:**

עמוד זה מציג מידע מפורט על העסק הרשום במערכת NexTor. מטרת העמוד היא לאפשר לבעל העסק לצפות בפרטים חשובים על העסק, כולל שעות פעילות, אמצעי התקשרות ותמונות ייצוגיות.

**תוכן העמוד:**

**פרטי העסק:** שם העסק, כתובת העסק, מספר טלפון ליצירת קשר ושעות פעילות העסק.

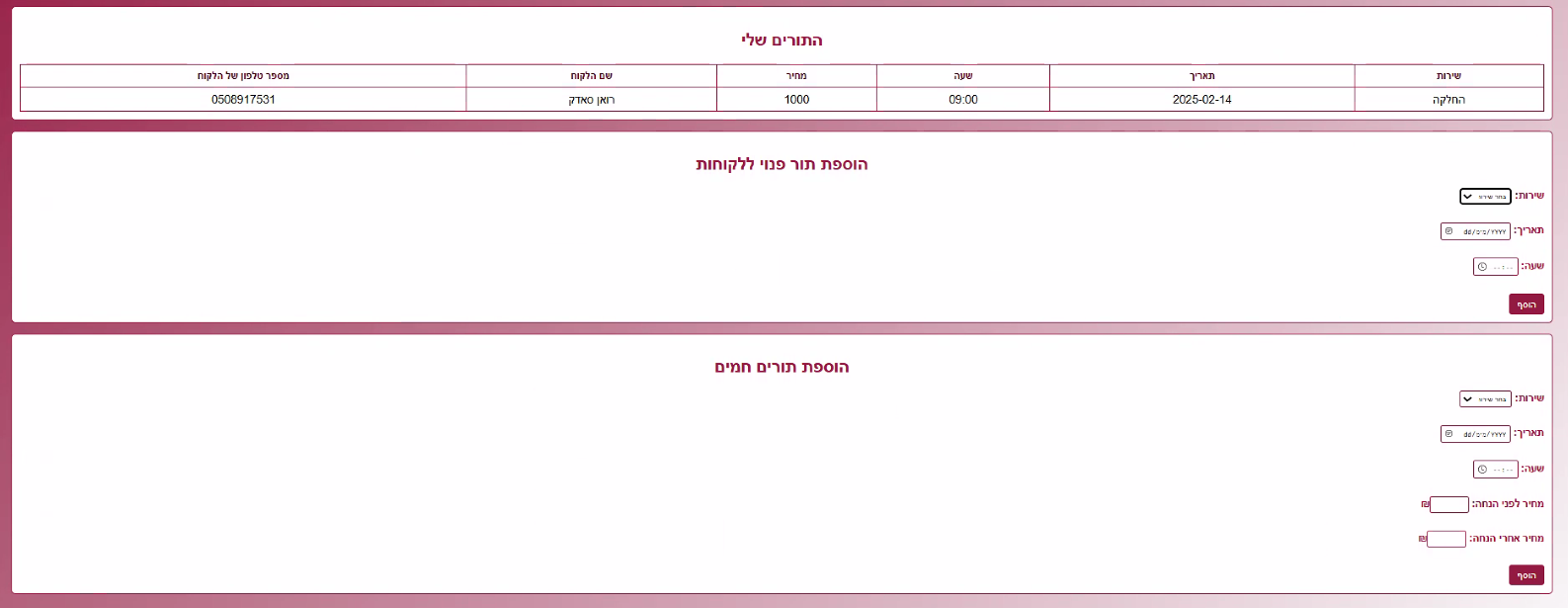
**תמונות העסק:** גלריה של תמונות ייצוגיות המציגות את העסק והשירותים שהוא מציע.

**כפתור עריכת פרטים:** מאפשר לבעלי העסק לעדכן את פרטי העסק, כולל כתובת, שעות פעילות ותמונות.

**הערות למשתמש:**

יש לוודא שפרטי העסק המוצגים מדויקים ומעודכנים.

ניתן לפנות לתמיכה במקרה של שאלות או בעיות טכניות דרך עמוד "צור קשר".

**ניהול תורים עמוד בעל העסק:**

עמוד זה מאפשר לבעלי עסקים לנהל את התורים הקיימים שלהם, להוסיף תורים חדשים, ולנהל תורים חמים במערכת NexTor.

**תוכן העמוד:**

**התורים שלי:** טבלה המציגה את התורים הקיימים עם הפרטים הבאים: שם הלקוח, מספר טלפון של הלקוח, סוג השירות, תאריך ושעה, משך השירות ,מחיר השירות

וכפתור אישור וביטול תור שמאפשר לאשר או לבטל תור קיים.

**הוספת תור פנוי ללקוחות:** טופס המאפשר להוסיף תור חדש: שירות, תאריך ,שעה

וכפתור "הוסף" שמוסיף את התור החדש למערכת.

**הוספת תורים חמים:** טופס המאפשר להוסיף תורים חמים: שירות, תאריך ,שעה

וכפתור "הוסף" שמוסיף את התור החם למערכת.

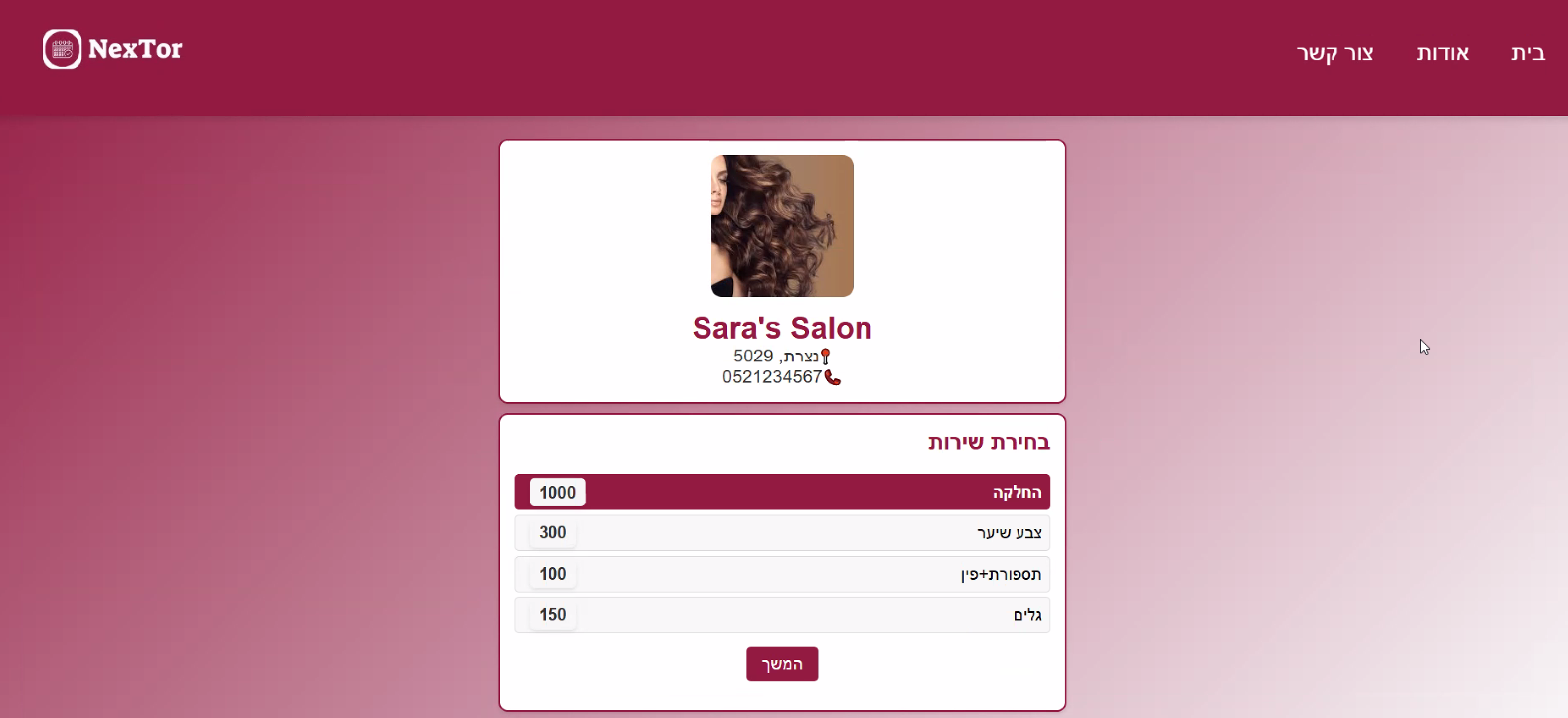
**הערות למשתמש:**

יש לוודא שכל פרטי התורים המוזנים מדויקים כדי למנוע טעויות.

מומלץ להוסיף תורים חמים עם הנחות משמעותיות כדי למשוך לקוחות.

במקרה של שאלות או בעיות, ניתן לפנות לעמוד "צור קשר".

**Booking screen:**

**עמוד זימון תור לעסק מסוים:**

עמוד זה מציג ללקוחות את פרטי העסק ואת השירותים שהוא מציע. העמוד מאפשר ללקוחות לבחור שירותים להזמנה בצורה קלה ונוחה.

**תוכן העמוד:**

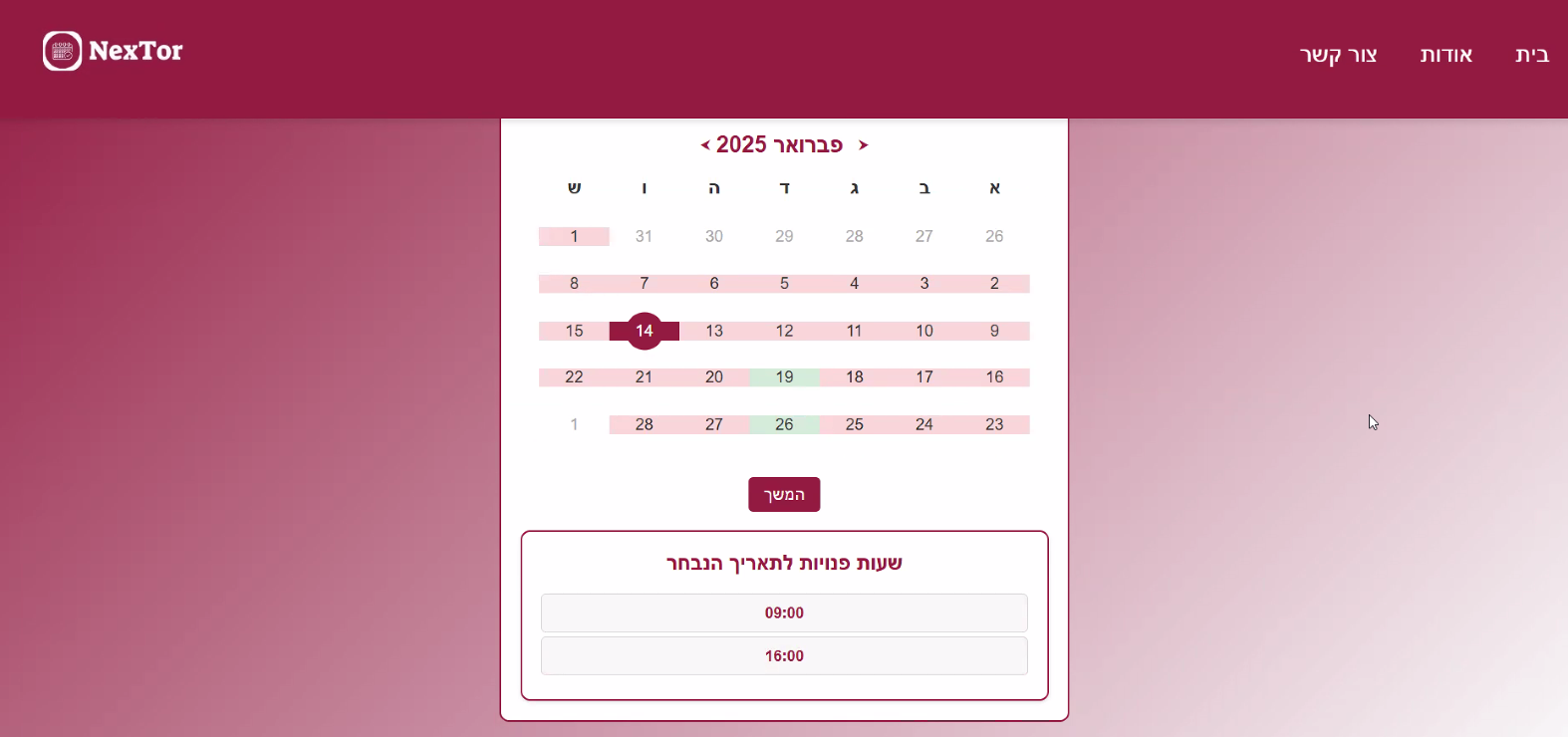
**פרטי העסק:** שם העסק, כתובת העסק ומספר טלפון ליצירת קשר.

**רשימת שירותים לבחירה:** טבלה או רשימה המציגה את שמות השירותים המוצעים בעסק ומחיר כל שירות.

**הערות למשתמש:**

ניתן ללחוץ על השירות הרצוי כדי לבחור תאריך מתאים.

במקרה של שאלות, ניתן לפנות לבעל העסק דרך מספר הטלפון המופיע בעמוד.

**המשך עמוד זימון תור לעסק מסוים - אחרי בחירת שירות:**

עמוד זה מאפשר למשתמשים לבחור תאריך ושעה להזמנת שירות שבחרו דרך המערכת של NexTor.

**תוכן העמוד:**

**לוח שנה:** להציג תאריכים פנויים ואפשריים לבחירת תור, ימים פנויים מסומנים בצורה ברורה. יש אפשרות לניווט בין חודשים בלחיצה על חיצים קדימה ואחורה.

**רשימת שעות זמינות:** לאחר בחירת התאריך, מוצגות השעות הפנויות עבור אותו יום, כל שעה פנויה מוצגת כלחצן לבחירה. לחיצה על השעה הרצויה תאפשר המשך בתהליך הזמנת התור.

**כפתור "המשך":** משמש לאישור התאריך והשעה שנבחרו ולעבור לשלב הבא בתהליך ההזמנה.

**הערות למשתמש:**

יש לבחור תאריך ושעה פנויים בלבד כפי שמסומנים בלוח השנה.

במקרה של אי זמינות, ניתן לנסות תאריך אחר או להירשם ברשימת המתנה של תורים לא זמינים.

**תמונה שמכילה טקסט, צילום מסך, עיצוב

התיאור נוצר באופן אוטומטיהמשך עמוד זימון תור לעסק מסוים - אחרי בחירת תאריך ושעה:**

עמוד זה מאפשר למשתמשים להזין את פרטיהם האישיים להשלמת הזמנת השירות ולמעבר לתהליך התשלום.

**תוכן העמוד:**

**טופס להזנת פרטי המשתמש:** שם מלא, מספר טלפון והערות שהוא שדה הזנה חופשי למילוי הערות או בקשות מיוחדות הקשורות לתור.

**כפתור "המשך לתשלום":** לאחר מילוי כל השדות, לחיצה על כפתור זה תעביר את המשתמש לעמוד התשלום.

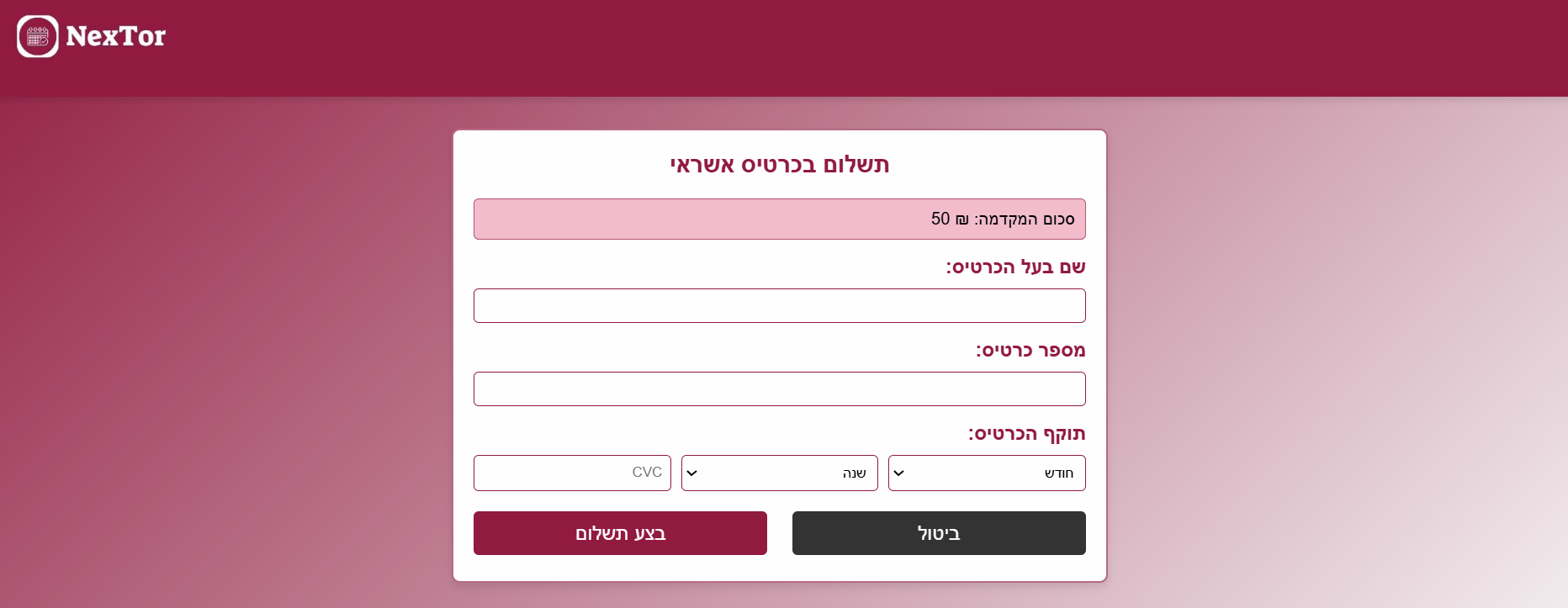
**הערות למשתמש:**

יש למלא את כל השדות בצורה נכונה ומלאה כדי להבטיח את תקינות ההזמנה.

ניתן להשתמש בשדה "הערות" להוספת בקשות מיוחדות או מידע נוסף הקשור להזמנה.

במקרה של בעיות במילוי הטופס, ניתן לפנות לתמיכה דרך עמוד "צור קשר".

**Payment screen:**



**עמוד תשלום בכרטיס אשראי:**

עמוד זה מאפשר למשתמשים לבצע תשלום מקדמה עבור השירותים שהוזמנו דרך אתר NexTor.

**שדות ותכונות בעמוד:**

**סכום המקדמה:** מצוין בראש העמוד (לדוגמה: "סכום המקדמה: 50 ₪").

**שם בעל הכרטיס:** המשתמש מזין את שמו כפי שמופיע על כרטיס האשראי.

**מספר כרטיס:** המשתמש מזין את מספר כרטיס האשראי שברשותו.

**תוקף הכרטיס:** המשתמש בוחר את החודש והשנה שבהם כרטיס האשראי בתוקף.

**שדות:** שדה לבחירת החודש ושדה לבחירת השנה.

**CVC (קוד אבטחה):** המשתמש מזין את שלושת הספרות המופיעות בגב הכרטיס.

**כפתור "בצע תשלום":** לאחר מילוי כל הפרטים, המשתמש לוחץ על הכפתור להשלמת התשלום.

**כפתור "ביטול":** מאפשר למשתמש לבטל את הפעולה ולחזור לעמוד הקודם ללא ביצוע התשלום.

**הערות למשתמש:**

יש למלא את כל השדות במדויק כדי להבטיח שהתשלום יבוצע בהצלחה.

במקרה של שגיאה או בעיה בתהליך התשלום, ניתן לפנות לתמיכה דרך עמוד "צור קשר".

חשוב לוודא שהתשלום מתבצע בסביבה מאובטחת, כפי שמוצג בעמוד זה.

**My appointments screen:**

**תמונה שמכילה טקסט, צילום מסך, תוכנה, מערכת הפעלה

התיאור נוצר באופן אוטומטיעמוד התורים שלי:**

עמוד זה מאפשר למשתמשים לצפות בפרטי התורים שהוזמנו, לבצע שינויים או לבטל את התורים במידת הצורך.

**רשימת תורים אישיים שכוללת:**

שם העסק.

כתובת העסק.

סוג השירות שהוזמן.

מחיר השירות.

תאריך ושעה של התור.

משך השירות.

סטטוס התור (לדוגמה: תור פעיל או תור פסול).

**כפתורים לפעולות בתור:**

**ביטול תור:** מאפשר לבטל תור שכבר נקבע. לאחר לחיצה, המערכת תבקש אישור לפני השלמת הביטול.

**העבר לחבר:** מאפשר להעביר את התור ללקוח אחר במערכת ע"י העתקת קישור.

**הערות למשתמש:**

במקרה של בעיות טכניות, ניתן לפנות לעמוד "צור קשר" לסיוע נוסף.

**תמונה שמכילה טקסט, צילום מסך, תוכנה, עיצוב

התיאור נוצר באופן אוטומטיעמוד התורים שלי - תורים ברשימת המתנה:**

עמוד זה מאפשר למשתמשים לצפות בפרטי התור שנמצא ברשימת ההמתנה שלהם ולבצע פעולות ניהוליות כמו מחיקת התור במידת הצורך.

**כרטיס תור ברשימת ההמתנה:**

שם העסק.

כתובת העסק.

סוג השירות.

תאריך ושעה שנקבעו לתור.

**כפתור מחיקה מרשימת ההמתנה:**

מאפשר למשתמש למחוק את התור מרשימת ההמתנה. בלחיצה על הכפתור, המערכת תבקש אישור למחיקת התור לפני השלמת הפעולה.

**הערות למשתמש:**

מחיקת תור מרשימת ההמתנה תגרום להסרתו מהמערכת ולא תהיה אפשרות לשחזר אותו.

למידע נוסף או סיוע טכני, ניתן לפנות לעמוד "צור קשר".

**Maintenance Guide**

This guide is intended to enable the continued use, maintenance, and enhancement of the NexTor project. It includes instructions for implementing changes, updates, and improvements to ensure the system remains operational and relevant.

**The website URL:**

<https://nex-tor.onrender.com>

**Operating Environment:**

The NexTor system requires the following software for proper functioning:

* Any modern browser (Google Chrome, Mozilla Firefox, Microsoft Edge).
* Code Editor: Visual Studio Code or any preferred text editor.
* Node.js for running JavaScript-related backend tasks.
* Web Server.
* MongoDB for database.

**Frontend - Components overview:**

**CSS -** contains the styles for all pages**:**

**about.css:** Styles for the about page.

**addBusiness.css:** Styles for adding a business.

**booking.css:** Styles for the booking page.

**business.css :** Styles for displaying businesses.

**contactUs.css:** Styles for the contact us page.

**forgetPass.css:** Styles for the forgot password page.

**home.css:** Styles for the homepage.

**login.css:** Styles for the login page.

**myAppointments.css:** Styles for managing user appointments.

**payment.css:** Styles for the payment page.

**signup.css:** Styles for the signup page.

**Images –** contains all the images in the project**:**

**copy.jpg:** copy image.

**logo.jpg:** NexTor logo.

**JavaScript -** contains specific logic and provides interactive functionalities **:**

**about.js:** Handles functionality for the about page.

**addBusiness.js:** Logic for adding a business.

**booking.js:** Handles booking related functionalities.

**business.js:** Functions for displaying and managing business data.

**contactUs.js:** Handles contact form submission.

**forgetPass.js:** Logic for password recovery.

**home.js:** Scripts for the homepage features.

**login.js:** Handles user authentication.

**myAppointments.js:** Manages user appointments.

**payment.js:** Handles payment processing.

**signup.js:** Manages user registration.

**Pages -** contains the structure and content for each page**:**

**about.html:** About page structure.

**addBusiness.html:** Interface for adding a business.

**booking.html:** Page for making appointments.

**business.html:** Displays business-related information.

**contactUs.html:** Contact form structure.

**forgetPass.html:** Password recovery page.

**home.html:** Landing page.

**login.html:** User login page.

**myAppointments.html:** User appointments page.

**payment.html:** Payment interface.

**signup.html:** User registration page.

**index.html:** Main entry point.

**Backend - Components overview:**

**Config:**

**database.js:** Contains database connection logic for MongoDB.

**multerConfig.js:** Handles file upload configurations using Multer.

**Controllers:**

**addBusinessController.js:** Handles logic for adding a new business.

**authController.js:** Handles user authentication and session management.

**bookingController.js:** Manages appointment bookings.

**businessController.js:** Logic for managing business-related data.

**forgotPasswordController.js:** Handles password recovery processes.

**homeController.js:** Provides functionality for the home page.

**myAppointmentsController.js:** Manages user appointments.

**paymentController**.js: Manages advance payment process.

**Middleware:**

**authMiddleware.js:** Provides middleware for authentication and access control.

**Models:**

**Appointment.js:** Defines the schema for appointments.

**businessModel.js:** Defines the schema for business entities.

**User.js:** Defines the schema for user accounts.

**Routes:**

**addBusinessRoutes.js:** Defines endpoints for adding a business.

**authRoutes.js:** Contains authentication-related routes.

**bookingRoutes.js**: Routes for booking appointments.

**businessRoutes.js:** Manages routes related to business data.

**forgotPasswordRoutes.js:** Routes for password recovery.

**homeRoutes.js:** Provides routes for the home page.

**myAppointmentsRoutes.js:** Routes for user appointment management.

**paymentRoutes.**js: Routes for payment management.