BrickEla Cuts Documentation

Table of Contents

User Documentation

1. Getting Started
    - System Requirements
    - Account Creation
    - Email Verification
    - Logging In
2. Customer Features
    - Booking an Appointment
    - Managing Appointments
    - Canceling Appointments
    - Profile Management
3. Barber Features
    - Accessing the Barber Dashboard
    - Viewing Appointments
    - Managing Schedule
    - Handling Customer Requests
4. Administrator Features
    - Admin Dashboard Overview
    - Managing Barbers
    - Service Management
    - User Management
    - System Settings

Developer Documentation

1. System Architecture
    - Technology Stack
    - Directory Structure
    - Database Schema
2. Setup and Installation
    - Development Environment
    - Dependencies
    - Configuration
    - Database Setup
3. Frontend Development
    - Component Structure
    - State Management
    - Routing System
    - API Integration

- Styling Guidelines

4. Backend Development
   - API Endpoints
   - Authentication System
   - Email Service
   - Database Operations
   - Security Measures
5. Deployment
   - Production Build
   - Server Configuration
   - SSL Setup
   - Monitoring

User Documentation

Getting Started

System Requirements

- Modern web browser (Chrome, Firefox, Safari, Edge)
- Stable internet connection
- Email account for registration
- Mobile or desktop device

Account Creation

1. Navigate to the registration page
2. Fill in required information:
   - First Name
   - Last Name
   - Email Address
   - Phone Number
   - Password (minimum 8 characters)
3. Accept terms and conditions
4. Click "Register"

Email Verification

1. Check your email inbox for verification link
2. Click the verification link
3. Wait for confirmation message
4. Proceed to login

Logging In

1. Enter your email address
2. Enter your password
3. Click "Login"
4. Use "Forgot Password" if needed

Customer Features

Booking an Appointment

1. Navigate to "Book Appointment"
2. Select desired service
3. Choose preferred barber
4. Select date and time
5. Add any special notes
6. Confirm booking
7. Check email for confirmation

Managing Appointments

1. View upcoming appointments
2. Check appointment details
3. Receive email reminders
4. View appointment history

Canceling Appointments

1. Locate appointment in dashboard
2. Click "Cancel Appointment"
3. Confirm cancellation
4. Note: 24-hour cancellation policy applies

Profile Management

1. Update personal information
2. Change password
3. Manage notification preferences
4. View booking history

Barber Features

Accessing the Barber Dashboard

1. Login with barber credentials
2. Navigate to barber dashboard
3. View daily schedule

Viewing Appointments

1. Check daily/weekly schedule
2. View customer details
3. Check service details
4. Manage appointment status

Managing Schedule

1. Set working hours
2. Mark unavailable times
3. View upcoming appointments
4. Handle schedule conflicts

Administrator Features

Admin Dashboard Overview

1. Access admin panel
2. View system statistics
3. Monitor bookings

4. Generate reports

Managing Barbers

1. Add new barbers
2. Edit barber profiles
3. Set working hours
4. Manage permissions
5. View performance metrics

Service Management

1. Add new services
2. Edit existing services
3. Set pricing
4. Manage service duration
5. Configure availability

Developer Documentation

System Architecture

Technology Stack

Frontend:

- React 18.3.1
- TypeScript
- TailwindCSS
- Framer Motion
- Axios for API communication

Backend:

- PHP 7.4+
- MySQL Database
- JWT Authentication
- PHPMailer

Directory Structure

```
project/
├── src/
│   ├── api/
│   │   ├── admin/
│   │   ├── services/
│   │   ├── middleware/
│   │   └── config.php
│   ├── components/
│   │   ├── admin/
│   │   ├── barber/
│   │   └── common/
│   ├── context/
│   ├── hooks/
│   └── types/
```

```
├── public/
└── docs/
```

Database Schema

```
Tables:
- ugyfelek (Users)
- fodraszok (Barbers)
- szolgaltatasok (Services)
- foglalasok (Bookings)
- rate_limits
```

Setup and Installation

Development Environment

1. Required software:
   - Node.js 14+
   - PHP 7.4+
   - MySQL 5.7+
   - Composer
   - Git
2. IDE Setup:
   - VS Code recommended
   - PHP extensions
   - ESLint configuration
   - Prettier setup

Frontend Development

Component Structure

- Atomic design principles
- Reusable components
- Props interface definitions
- Component documentation

State Management

- Context API usage
- Custom hooks
- State organization
- Data flow patterns

API Integration

```
// Example API service
export const createBooking = async (
  bookingData: BookingData
): Promise<void> => {
  try {
    const response = await apiClient.post<ApiResponse<void>>(
      "/create-booking.php",
      bookingData
    );
    if (!response.data.success) {
      throw new ApiError(response.data.message || "Booking failed");
    }
  } catch (error) {
    handleApiError(error);
  }
};
```

Backend Development

API Endpoints

Authentication

```
POST /api/login.php
POST /api/register.php
GET /api/verify-email.php
```

Bookings

```
POST /api/create-booking.php
GET /api/get-appointments.php
POST /api/cancel-booking.php
```

Admin

```
GET /api/admin/dashboard-stats.php
POST /api/admin/add-barber.php
POST /api/admin/add-service.php
```

Security Measures

1. Input Validation

```
// Example input validation
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    throw new Exception('Invalid email format');
}
```

2. Rate Limiting

```
// Example rate limiting check
$stmt = $conn->prepare("SELECT COUNT(*) as attempts
    FROM rate_limits
    WHERE ip_address = ?
    AND action = 'login_failed'
    AND timestamp > DATE_SUB(NOW(), INTERVAL 1 HOUR)");
```

3. JWT Authentication

```
// Example JWT token generation
$token = JWT::encode([
    'user_id' => $user['UgyfelID'],
    'email' => $user['Email'],
    'exp' => time() + (60 * 60 * 24)
], JWT_SECRET_KEY, 'HS256');
```

Deployment

Production Build

1. Frontend Build

```
npm run build
```

2. Backend Deployment

- Configure PHP environment
- Set up MySQL database
- Configure email service

Server Configuration

1. Apache/Nginx setup
2. PHP configuration
3. MySQL optimization
4. SSL certificate installation

Monitoring

1. Error logging
2. Performance monitoring
3. Security auditing
4. Backup systems

Support and Maintenance

Troubleshooting

1. Common Issues
2. Error Messages
3. Resolution Steps

Updates and Maintenance

1. Version Control
2. Database Backups
3. Security Updates
4. Performance Optimization