

Guest Reservation Portal for Casa Tente Boutique Hotel – Technical Manual

1. SYSTEM OVERVIEW

The **Guest Reservation Portal** is a web-based hotel management system designed for Casa Tente Boutique Hotel in Matnog, Sorsogon. It replaces manual booking processes with an automated, secure, and user-friendly platform.

1.1 System Components

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Node.js
- **Database:** MySQL
- **Hosting:** Supabase
- **Development Environment:** Visual Studio Code

1.2 User Roles

1. **Customer:** Books rooms, views amenities, submits feedback
2. **Administrator:** Manages rooms, reservations, amenities, feedback, and analytics

2. INSTALLATION GUIDE (FOR LOCAL DEVELOPMENT)

2.1 Prerequisites

Hardware Requirements (Minimum)

- **Memory:** 8GB RAM
- **Processor:** Intel Core i3 13th Gen or equivalent
- **Storage:** 20GB available space
- **Internet:** 20 Mbps connection

Software Requirements

- **Operating System:** Windows 10/11 (64-bit)
- **Browser:** Google Chrome v.134 or later

- **Development Tools:**

- Visual Studio Code v.1.106.0 or later
- Node.js (Latest LTS version)
- MySQL v.10.4.32-MariaDB (via XAMPP)
- Git (for version control)

2.2 Step-by-Step Installation

Step 1: Environment Setup

1. Install Node.js

- Download from <https://nodejs.org>
- Run installer and follow prompts
- Verify installation:

```
bash
```

```
node --version
```

```
npm --version
```

2. Install XAMPP

- Download from <https://www.apachefriends.org>
- Install with default settings
- Start Apache and MySQL from XAMPP Control Panel

3. Install Visual Studio Code

- Download from <https://code.visualstudio.com>
- Install with recommended settings

Step 2: Database Setup

1. Access phpMyAdmin

- Open browser and navigate to: <http://localhost/phpmyadmin>

2. Create Database

```
sql
```

```
CREATE DATABASE casatente_hotel;
```

USE casatente_hotel;

3. Import Database Schema

- Use the provided SQL file casatente_database.sql
- Import through phpMyAdmin import feature

Step 3: Project Setup

1. Clone/Download Project Files

bash

git clone [repository-url]

cd guest-reservation-portal

2. Install Dependencies

bash

npm install

3. Configure Environment Variables

Create .env file in root directory:

env

DB_HOST=localhost

DB_USER=root

DB_PASSWORD=

DB_NAME=casatente_hotel

PORT=3000

EMAIL_HOST=smtp.gmail.com

EMAIL_PORT=587

EMAIL_USER=hotelecastenteboutique@gmail.com

EMAIL_PASS=[your-email-password]

Step 4: Run the Application

1. Start Development Server

bash

npm start

or

bash

node app.js

2. Access Application

- Open browser and navigate to: <http://localhost:3000>
- For admin: <http://localhost:3000/admin>

3. DEPLOYMENT TO SUPABASE

3.1 Supabase Setup

1. Create Supabase Account

- Visit <https://supabase.com>
- Sign up and create new project

2. Configure Database

- Navigate to SQL Editor
- Import database schema
- Set up tables: users, rooms, reservations, amenities, feedback

3. Environment Configuration

Update .env file:

env

SUPABASE_URL=[your-supabase-url]

SUPABASE_KEY=[your-supabase-anon-key]

SUPABASE_SERVICE_KEY=[your-supabase-service-key]

3.2 Deploy Application

1. Build for Production

bash

npm run build

2. Deploy to Hosting Service

- Options: Vercel, Heroku, or DigitalOcean
- Connect GitHub repository
- Configure environment variables
- Deploy application

3. Verify Deployment

- Test all functionalities
- Check email notifications
- Verify database connections

4. SYSTEM NAVIGATION GUIDE

4.1 Customer Interface

4.1.1 Home Page

- **URL:** http://localhost:3000 or your deployed URL
- **Features:**
 - View hotel branding and tagline
 - Navigation menu: Home, Rooms, Amenities, About, Contact, Login
 - Hero image showcasing hotel

4.1.2 Account Creation

1. **Click "Sign Up"** from Login page
2. **Fill registration form:**
 - Name, Surname, Email, Password
 - Confirm password
3. **Receive OTP** via email
4. **Verify account** and login

4.1.3 Room Browsing

1. **Navigate to "Rooms"**

2. **View room categories:**

- **Standard Rooms:** ₱500/night, 2 guests max
- **Deluxe Rooms:** ₱2,500/night, 4 guests max

3. **Click "View Details"** for room specifications

4.1.4 Making Reservations

1. **Select room** and click "Book Now"
2. **Login required** if not already logged in
3. **Fill reservation form:**
 - Personal information
 - Check-in/Check-out dates (calendar picker)
 - Number of guests (max 4)
 - Special requests
4. **Submit reservation** → Receive confirmation email with PDF slip

4.1.5 Feedback Submission

1. **Navigate to "Contact"** or feedback section
2. **Login required**
3. **Submit rating** (1-5 stars) and comments
4. **Receive confirmation** of submission

4.2 Administrator Interface

4.2.1 Admin Login

- **URL:** http://localhost:3000/admin
- **Credentials:** Provided by system administrator
- **Dashboard:** Overview of key metrics

4.2.2 Rooms Management

Path: Admin → Rooms

- **View all rooms** with ID, Name, Type, Price, Status
- **Actions:**

- **Edit:** Update room details, amenities, images
- **Delete:** Remove room (with confirmation)
- **Add new rooms** via "Add Room" button

4.2.3 Reservations Management

Path: Admin → Reservations

- **View all bookings** in table format:
 - ID, Room, Guest, Dates, Total, Status
- **Status Management:**
 - **Pending** → Click "Confirm"
 - **Confirmed** → Click "Cancel" if needed
 - **In Progress** → Currently occupied
 - **Completed** → Stay finished
 - **Cancelled** → Booking cancelled
- **Calendar View:** Visual schedule of all reservations

4.2.4 Amenities Management

Path: Admin → Amenities

- **Two categories:**
 1. **Indoor Amenities:** AC, Mini Refrigerator, etc.
 2. **Outdoor Amenities:** Pool, Function Hall, Parking
- **Actions:** Edit descriptions, upload images

4.2.5 Feedback Management

Path: Admin → Feedbacks

- **View all customer reviews**
- **Filter by rating** (1-5 stars)
- **Monitor satisfaction** trends

4.2.6 Analytics Dashboard

Path: Admin → Analytics

- **Key Metrics:**
 - Total Revenue
 - Revenue This Month
 - Total Reservations
 - Reservations This Month
 - Average Rating
 - **Visual Charts:**
 - Revenue Trend (Line graph)
 - Reservation Status Distribution (Pie chart)
 - **Insights:**
 - Most Reserved Room
 - Most Frequent Customer
-

5. SYSTEM CONFIGURATION

5.1 Email Configuration

1. **SMTP Settings** for booking confirmations:

javascript

// In email configuration file

```
const transporter = nodemailer.createTransport({  
  host: 'smtp.gmail.com',  
  port: 587,  
  secure: false,  
  auth: {  
    user: 'hotelecastenteboutique@gmail.com',  
    pass: 'your-app-password'  
  }  
});
```


2. Email Templates:

- Registration OTP
- Reservation confirmation with PDF slip
- Status updates

5.2 Database Configuration

javascript

// Database connection setup

```
const pool = mysql.createPool({  
  host: process.env.DB_HOST,  
  user: process.env.DB_USER,  
  password: process.env.DB_PASSWORD,  
  database: process.env.DB_NAME,  
  connectionLimit: 10  
});
```

5.3 Security Configuration

1. **Password Encryption:** bcrypt hashing
2. **Session Management:** express-session with secure settings
3. **Input Validation:** sanitize all user inputs
4. **CORS Configuration:** Restrict to trusted domains

6. TROUBLESHOOTING

6.1 Common Issues

Issue 1: Database Connection Failed

Symptoms: "Cannot connect to database" error

Solutions:

1. Check if MySQL service is running
2. Verify database credentials in .env

3. Ensure database casatente_hotel exists
4. Check firewall settings

Issue 2: Email Not Sending

Symptoms: No confirmation emails received

Solutions:

1. Verify SMTP settings in .env
2. Check Gmail "Less secure apps" setting
3. Verify email account has proper permissions
4. Check spam folder

Issue 3: Page Loading Errors

Symptoms: Blank pages or 404 errors

Solutions:

1. Clear browser cache (Ctrl+Shift+Delete)
2. Check console for JavaScript errors
3. Verify all dependencies are installed
4. Restart Node.js server

Issue 4: File Upload Issues

Symptoms: Room/Amenity images not uploading

Solutions:

1. Check file size limits (max 2MB)
2. Verify file types allowed (jpg, png, webp)
3. Check upload directory permissions
4. Ensure sufficient disk space

6.2 Maintenance Procedures

Daily Tasks:

1. **Backup Database:**

bash

```
mysqldump -u root -p casatente_hotel > backup_$(date +%Y%m%d).sql
```

2. **Check Server Logs** for errors
3. **Monitor Performance** metrics
4. **Verify Email Queue** processing

Weekly Tasks:

1. **Update Dependencies:**

bash

npm update

2. **Clean Temporary Files**
3. **Review Error Logs**
4. **Test Backup Restoration**

Monthly Tasks:

1. **Security Audit**
2. **Performance Optimization**
3. **Database Optimization**
4. **Review User Feedback**

7. USER MANAGEMENT

7.1 Creating Admin Accounts

1. **Access Database Directly:**

sql

INSERT INTO users (name, email, password, role)

VALUES ('Admin Name', 'admin@email.com', '[hashed-password]', 'admin');

2. **Or use registration form** with admin privileges

7.2 Password Reset

1. **Customers:** Use "Forgot Password" feature
2. **Admins:** Database reset required

sql

UPDATE users SET password = '[new-hashed-password]' WHERE email = 'admin@email.com';

7.3 Role Management

- **Customer:** Can book, view, submit feedback
 - **Admin:** Full system access
 - **Future roles:** Manager, Staff (with limited access)
-

8. DATA MANAGEMENT

8.1 Exporting Reports

1. From Analytics Dashboard:

- Click "Export" button
- Choose format: CSV, Excel, PDF
- Select date range

2. Database Export:

bash

Export reservations

```
SELECT * FROM reservations WHERE check_in BETWEEN '2025-01-01' AND '2025-12-31';
```

Export to CSV

```
mysql -u root -p -e "SELECT * FROM reservations" casatente_hotel > reservations.csv
```

8.2 Data Backup Strategy

1. Automated Daily Backup (cron job):

bash

Add to crontab

```
0 2 * * * /path/to/backup-script.sh
```

2. Backup Script (backup-script.sh):

bash

```
#!/bin/bash
```

```
DATE=$(date +%Y%m%d_%H%M%S)
```

```
mysqldump -u root -p[password] casatente_hotel > /backups/casatente_$(DATE).sql
```

```
gzip /backups/casatente_$(DATE).sql
```

```
# Keep only last 30 days
```

```
find /backups -name "*.gz" -mtime +30 -delete
```

8.3 Data Privacy Compliance

1. Personal Data Protection:

- Encrypt sensitive information
- Regular data purging (after 2 years)
- Access logs for audit trail

2. GDPR Compliance:

- Right to access data
 - Right to deletion
 - Data portability option
-

9. SYSTEM MONITORING

9.1 Performance Monitoring

1. Response Time Tracking:

- Page load < 3 seconds
- Database queries < 100ms
- API response < 500ms

2. Uptime Monitoring:

- Target: 99.9% uptime
- Monitor: <https://uptimerobot.com> or similar

9.2 Error Monitoring

1. Application Logs:

javascript

// Log configuration

```
const winston = require('winston');  
  
const logger = winston.createLogger({  
  level: 'info',  
  format: winston.format.json(),  
  transports: [  
    new winston.transports.File({ filename: 'error.log', level: 'error' }),  
    new winston.transports.File({ filename: 'combined.log' })  
  ]  
});
```

2. Error Alerting:

- Email notifications for critical errors
- SMS alerts for system downtime

10. SUPPORT AND CONTACT

10.1 Technical Support

- **Email:** npcolumna@sorsu.edu.ph
- **Development Team:**
 - Neil Patrick L. Columna (Programmer)
 - Rhenalie G. Villanueva (System Analyst/Technical Writer)
 - Sonlife M. Guray (Contributor)

10.2 Hotel Contact

- **Phone:** (052) 123-4567 | +63 912 345 6789
- **Email:** hotelecastenteboutique@gmail.com
- **Facebook:** <https://www.facebook.com/casatente>

10.3 Emergency Procedures

1. **System Down:** Contact development team immediately
 2. **Data Breach:** Isolate system, contact team, notify users
 3. **Hardware Failure:** Restore from latest backup
-