# ✈️ Flight Status Notification System

- Welcome to the Flight Status Notification System presentation!
- This system provides real-time flight status updates and customizable notifications.
- Developed using cutting-edge technologies to ensure scalability, reliability, and user satisfaction.

# 🏗️ System Architecture

## Backend:

Built with Node.js, Express, and Sequelize.

Utilizes PostgreSQL for robust data management.

Real-time data fetching from AviationStack API.

## Frontend:

Developed with React for a responsive and interactive user experience.

User interface allows managing flight notifications and preferences.

## Notification System:

RabbitMQ for efficient message queuing.

Twilio for SMS and WhatsApp notifications.

SendGrid for email notifications.

# 📈 Scalability and Robustness

**Scalable Design:**

- Scheduler-based architecture ensures efficient resource usage.
- RabbitMQ enables handling high volumes of notification messages.

**Robust Data Management:**

- PostgreSQL provides reliable and scalable data storage.
- Efficient indexing and querying for real-time performance.

**Modular Codebase:**

- Clean separation of concerns allows easy maintenance and scaling.
- Flexible to accommodate future enhancements and integrations.

# ✈️ Real-time Flight Data Integration

**AviationStack API:**

- Fetches real-time flight status updates.
- Provides essential data points like flight number, departure, and arrival times.

**Data Handling:**

- Only fetches data for registered users' flights.
- Ensures efficient API usage and reduces unnecessary data fetching.

**Future Enhancements:**

- Potential integration with more comprehensive flight data sources.
- Expansion to include additional data points as needed.

# 🛠 Customizable Notifications

**User Preferences:**

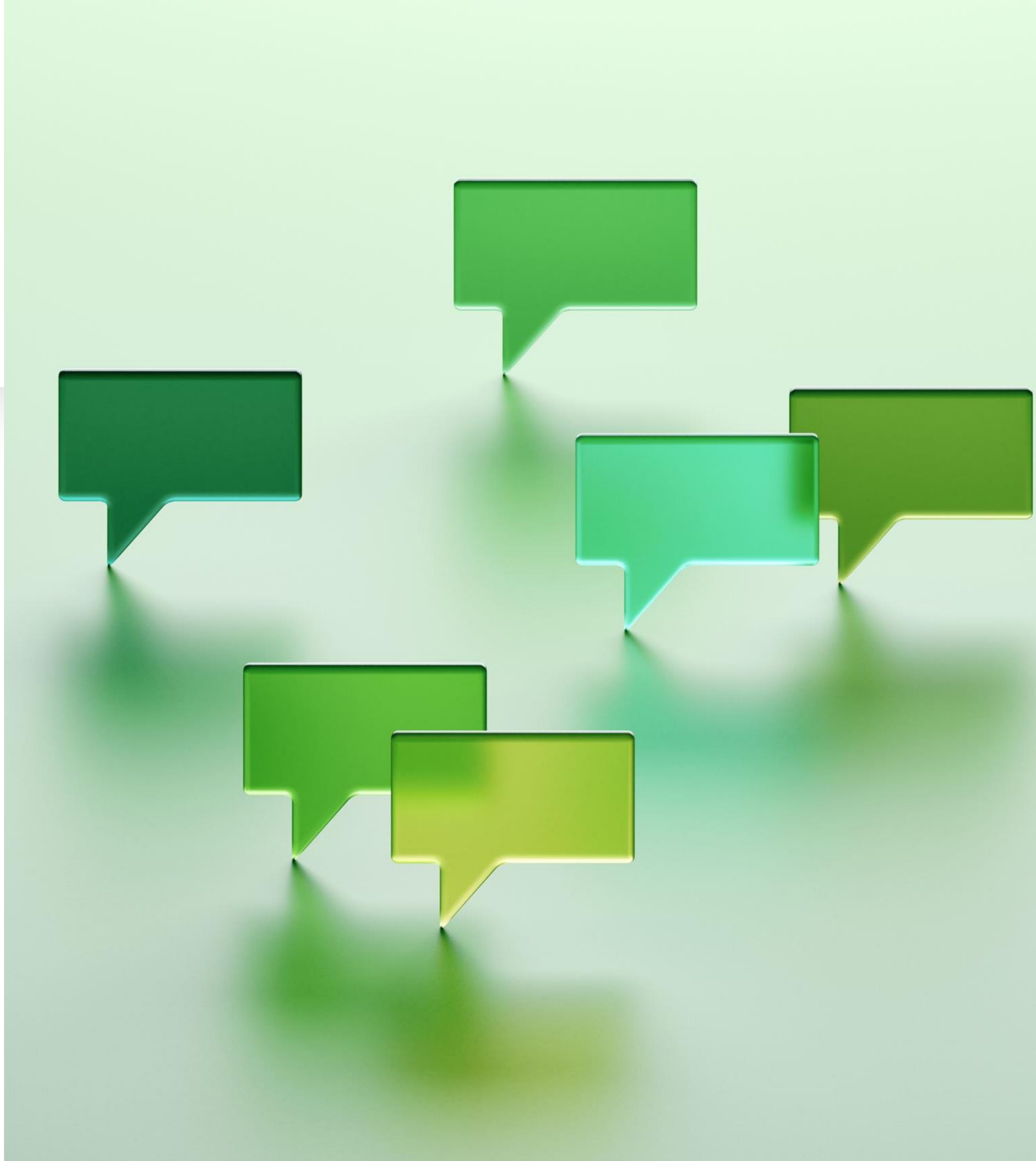
- Users can choose to receive notifications via SMS, Email, or WhatsApp.
- Easy-to-use interface for managing notification settings.

**Notification Delivery:**

- Twilio handles SMS and WhatsApp messages.
- SendGrid ensures reliable email delivery.

**Real-time Updates:**

- Notifications sent immediately upon flight status changes.
- Ensures users are always informed about their flight status.



# 🛠️ Technology Choices

**PostgreSQL:**

- Chosen for its robustness and scalability.
- Efficient data storage and retrieval for real-time performance.

**RabbitMQ:**

- Manages notification queues efficiently.
- Scalable to handle high volumes of messages.

**Twilio and SendGrid:**

- Reliable and widely-used services for SMS, WhatsApp, and Email.
- Ensure high delivery rates and real-time notifications.

# 🚀 Conclusion and Future Work

**Achievements:**

- Successfully developed a scalable and robust flight status notification system.
- Provided users with a customizable notification experience.

**Future Enhancements:**

- Integrate with more comprehensive flight data sources.
- Expand notification options and platforms.
- Further optimize performance and scalability.