

Here are the API endpoints that will be needed for the group travel calendar application, aligned with the functionalities described in your process flow:

## 1. User Onboarding and Authentication

### New User Registration

- **POST /api/auth/register**
  - Request Body: { "name": "string", "email": "string", "password": "string" }
  - Response: { "message": "Registration successful. Verification email sent." }
- **POST /api/auth/social-register**
  - Request Body: { "provider": "google|facebook", "accessToken": "string" }
  - Response: { "message": "Registration successful." }
- **POST /api/auth/verify-email**
  - Request Body: { "email": "string", "verificationCode": "string" }
  - Response: { "message": "Email verified successfully." }

### Existing User Login

- **POST /api/auth/login**
  - Request Body: { "email": "string", "password": "string" }
  - Response: { "token": "jwt-token", "user": { "id": "string", "name": "string" } }
- **POST /api/auth/social-login**
  - Request Body: { "provider": "google|facebook", "accessToken": "string" }
  - Response: { "token": "jwt-token", "user": { "id": "string", "name": "string" } }
- **POST /api/auth/forgot-password**
  - Request Body: { "email": "string" }
  - Response: { "message": "Password reset link sent." }

## 2. Calendar Integration

### Connect Calendar

- **POST /api/calendar/connect**
  - Request Body: { "provider": "google|outlook", "oauthToken": "string" }
  - Response: { "message": "Calendar connected successfully." }
- **GET /api/calendar/providers**
  - Response: { "providers": ["google", "outlook"] }

## 3. Group Creation and Management

## Create New Group

- **POST /api/groups/create**
  - Request Body: { "name": "string", "description": "string", "travelDates": "date-range" }
  - Response: { "groupId": "string", "invitationLink": "string" }

## Join Existing Group

- **POST /api/groups/join**
  - Request Body: { "groupId": "string", "invitationCode": "string" }
  - Response: { "message": "Joined group successfully." }

## 4. Availability Visualization

### Fetch Calendar Data

- **GET /api/calendar/events**
  - Request Params: ?groupId=string
  - Response: { "events": [ { "userId": "string", "events": [ ... ] } ] }

### Display Combined Availability

- **GET /api/calendar/availability**
  - Request Params: ?groupId=string&dateRange=optional
  - Response: { "availability": [ ... ] }

## 5. AI-Powered Scheduling Suggestions

### Analyze Availability

- **POST /api/schedule/analyze**
  - Request Body: { "groupId": "string", "preferences": { ... } }
  - Response: { "analysis": { "optimalSlots": [ ... ] } }

### Generate Suggestions

- **POST /api/schedule/suggest**
  - Request Body: { "groupId": "string", "preferences": { ... } }
  - Response: { "suggestions": [ ... ] }

## 6. Collaborative Itinerary Planning

### Shared Itinerary

- **POST /api/itinerary/create**
  - Request Body: { "groupId": "string", "itinerary": [ ... ] }
  - Response: { "itineraryId": "string", "message": "Itinerary created successfully." }
- **PUT /api/itinerary/update**
  - Request Body: { "itineraryId": "string", "updates": { ... } }
  - Response: { "message": "Itinerary updated successfully." }

## Communication and Decision-Making

- **POST /api/chat/send**
  - Request Body: { "groupId": "string", "message": "string" }
  - Response: { "messageId": "string", "timestamp": "datetime" }
- **POST /api/polls/create**
  - Request Body: { "groupId": "string", "question": "string", "options": [ ... ] }
  - Response: { "pollId": "string", "message": "Poll created successfully." }

## 7. Travel Booking (Optional)

### Integrate with Travel Platforms

- **POST /api/travel/search**
  - Request Body: { "groupId": "string", "criteria": { ... } }
  - Response: { "results": [ ... ] }
- **POST /api/travel/book**
  - Request Body: { "travelId": "string", "paymentInfo": { ... } }
  - Response: { "bookingId": "string", "message": "Booking confirmed." }

## 8. Notifications and Reminders

### Send Timely Notifications

- **POST /api/notifications/send**
  - Request Body: { "groupId": "string", "notification": { ... } }
  - Response: { "message": "Notification sent successfully." }

## 9. User Profile and Settings

### Manage Profile

- **PUT /api/user/profile**
  - Request Body: { "userId": "string", "profile": { ... } }
  - Response: { "message": "Profile updated successfully." }

## App Settings

- **PUT /api/user/settings**
  - Request Body: { "userId": "string", "settings": { ... } }
  - Response: { "message": "Settings updated successfully." }

## Additional Considerations

- **Error Handling:** Ensure that all endpoints return appropriate error messages for validation errors, unauthorized access, and other potential issues.
- **Security:** Implement JWT authentication for securing endpoints, and enforce role-based access control where necessary.

This structure should cover all the core functionalities for your group travel calendar application, with clear and consistent API endpoints for frontend-backend communication.