API Documentation

Table of Contents

User Routes

1. Sign Up
POST /api/users/signup
Request Body: { "name": "John Doe", "email": "john@example.com", "password": "password123" }
Response: { "user": { "id": "123", "name": "John Doe", "email": "john@example.com" }, "token": "jwt-token" }

2. Sign In
POST /api/users/signin
Request Body: { "email": "john@example.com", "password": "password123" }
Response: { "user": { "id": "123", "name": "John Doe", "email": "john@example.com" }, "token": "jwt-token" }

3. Verify Email
GET /api/users/verifyEmail/:token
Response: { "message": "Email verified successfully" }

4. Send Password Reset Email
POST /api/users/sendPasswordResetEmail
Request Body: { "email": "john@example.com" }
Response: { "message": "Password reset email sent" }

5. Reset Password
POST /api/users/resetPassword
Request Body: { "token": "reset-token", "newPassword": "newpassword123" }
Response: { "message": "Password reset successfully" }

6. Get User
GET /api/users/getUser/:userId
Headers: { "Authorization": "Bearer jwt-token" }
Response: { "user": { "id": "123", "name": "John Doe", "email": "john@example.com" } }

7. Search Users (Admin Only)
GET /api/users/searchUser
Headers: { "Authorization": "Bearer jwt-token" }
Query Params: { "name": "John", "exactMatch": "false", "likesCount": "10", "decksCount": "5", "joinedAfter": "2023-01-01", "role": "user" }
Response: { "users": [{ "id": "123", "name": "John Doe", "email": "john@example.com", "likesCount": 10, "decksCount": 5 }] }

Deck Routes

1. Create Deck
POST /api/decks
Request Body: { "title": "New Deck" }
Response: { "deck": { "id": "123", "title": "New Deck", "created_by": "user-id" } }

2. Update Deck
PUT /api/decks/:id
Request Body: { "title": "Updated Deck Title", "cards": ["card-id-1", "card-id-2"] }
Response: { "deck": { "id": "123", "title": "Updated Deck Title", "cards": ["card-id-1", "card-id-2"] } }

3. Delete Deck (Admin Only)
DELETE /api/decks/:id
Headers: { "Authorization": "Bearer jwt-token" }
Response: { "message": "Deck soft deleted successfully" }

4. Get Deck
GET /api/decks/:id
Response: { "deck": { "id": "123", "title": "Deck Title", "created_by": "user-id" } }

5. Get Public Decks
GET /api/decks/public
Response: { "decks": [{ "id": "123", "title": "Deck Title", "created_by": "user-id" }] }

6. Get Favorite Decks
GET /api/decks/favorites
Headers: { "Authorization": "Bearer jwt-token" }
Response: { "decks": [{ "id": "123", "title": "Deck Title", "created_by": "user-id" }] }

7. Toggle Favorite
POST /api/decks/:id/favorite
Headers: { "Authorization": "Bearer jwt-token" }
Response: { "deck": { "id": "123", "title": "Deck Title", "favorites": ["user-id"] } }

8. Add Card to Deck
POST /api/decks/:deckId/cards
Request Body: { "cardId": "card-id" }
Response: { "deck": { "id": "123", "title": "Deck Title", "cards": ["card-id"] } }

9. Search Decks
GET /api/decks/search
Query Params: { "title": "Deck Title", "exactMatch": "false", "cardsCount": "10", "favoritesCount": "5", "postedAfter": "2023-01-01" }
Response: { "decks": [{ "id": "123", "title": "Deck Title", "cardsCount": 10,

"favoritesCount": 5 }] }

10. Get User Decks
GET /api/decks/getUserDecks
Headers: { "Authorization": "Bearer jwt-token" }
Response: { "decks": [{ "id": "123", "title": "Deck Title", "created_by": "user-id" }] }

Card Routes

1. Create Card
POST /api/cards
Request Body: { "title": "New Card", "content": "Card content" }
Response: { "card": { "id": "123", "title": "New Card", "content": "Card content", "created_by": "user-id" } }

2. Update Card
PUT /api/cards/:id
Request Body: { "title": "Updated Card Title", "content": "Updated content" }
Response: { "card": { "id": "123", "title": "Updated Card Title", "content": "Updated content" } }

3. Delete Card
DELETE /api/cards/:id
Headers: { "Authorization": "Bearer jwt-token" }
Response: { "message": "Card deleted successfully" }

4. Get Card
GET /api/cards/:id
Response: { "card": { "id": "123", "title": "Card Title", "content": "Card content" } }

5. Get User Cards
GET /api/cards/getUserCards
Headers: { "Authorization": "Bearer jwt-token" }
Response: { "cards": [{ "id": "123", "title": "Card Title", "content": "Card content" }] }