

Complete Authentication & User API Endpoints

1 User Registration

Method: POST

Endpoint: /api/auth/register/

Description: Registers a new user.

Request Body:

```
{  
  "userName": "JohnDoe",  
  
  "userEmail": "johndoe@example.com",  
  
  "userPhone": "1234567890",  
  
  "userAddress": "123 Main St",  
  
  "password": "strongpassword"  
}
```

Response:

```
{  
  "message": "User registered successfully"  
}
```

2 User Login

Method: POST

Endpoint: /api/auth/login/

Description: Logs in a user and returns a JWT token.

Request Body:

```
{  
  "userEmail": "johndoe@example.com",  
  "password": "strongpassword"  
}
```

Response:

```
{
```

```
        "token": "jwt_token_here"  
    }  


---


```

3 Get Logged-in User Profile

Method: GET

Endpoint: /api/users/me/

Description: Retrieves the logged-in user's profile details.

Authentication: Requires JWT token in the Authorization header.

Request Header:

Authorization: Bearer <JWT_TOKEN>

Response:

```
{  
    "userName": "JohnDoe",  
    "userEmail": "johndoe@example.com",  
    "userPhone": "1234567890",
```

```
    "userAddress": "123 Main St"
```

```
}
```

4 Update User Profile

Method: PUT

Endpoint: /api/users/me/

Description: Updates the logged-in user's profile details.

Authentication: Requires JWT token in the Authorization header.

Request Body:

```
{
```

```
    "userName": "JohnDoeUpdated",
```

```
    "userPhone": "9876543210",
```

```
    "userAddress": "456 New St",
```

```
    "profilePicURL":
```

```
        "http://example.com/profilepic.jpg"
```

```
}
```

Response:

```
{  
  "message": "Profile updated successfully"  
}
```

5 Forgot Password

Method: POST

Endpoint: /api/auth/password/forgot/

Description: Requests a password reset by sending an OTP or reset link to the user's email.

Request Body:

```
{  
  "userEmail": "johndoe@example.com"  
}
```

Response:

```
{  
  "message": "Password reset link sent to email"  
}
```

6 Reset Password Using OTP/Token

Method: POST

Endpoint: /api/auth/password/reset/

Description: Resets the password using the OTP/token sent to the user's email.

Request Body:

```
{  
  "token": "otp_token_here",  
  "newPassword": "newstrongpassword"  
}
```

Response:

```
{  
  "message": "Password reset successfully"  
}
```

7 Send OTP for Verification

Method: POST

Endpoint: /api/auth/otp/send/

Description: Sends an OTP to the user's email for verification.

Request Body:

```
{  
  "userEmail": "johndoe@example.com"  
}
```

Response:

```
{  
  "message": "OTP sent to email"
```

```
}
```

8 Verify OTP

Method: POST

Endpoint: /api/auth/otp/verify/

Description: Verifies the OTP sent to the user's email.

Request Body:

```
{
```

```
    "userEmail": "johndoe@example.com",
```

```
    "otp": "123456"
```

```
}
```

Response:

```
{
```

```
    "message": "OTP verified successfully"
```

```
}
```

9 Change Password

Method: POST

Endpoint: /api/auth/password/change/

Description: Allows a logged-in user to change their password by providing the current password and a new password.

Authentication: Requires JWT token in the Authorization header.

Request Body:

```
{
```

```
  "currentPassword": "old_password_here",
```

```
  "newPassword": "new_password_here",
```

```
  "confirmPassword": "new_password_here"
```

```
}
```

Response (Success):

```
{
```

```
  "message": "Password changed successfully"
```

```
}
```

Response (Error - Current Password Incorrect):

```
{
```

```
    "error": "Current password is incorrect"
```

```
}
```

Response (Error - New Password Mismatch):

```
{
```

```
    "error": "New password and confirm password do  
    not match"
```

```
}
```

Response (Error - Password Does Not Meet Security Requirements):

```
{
```

```
"error": "New password does not meet security requirements"
```

```
}
```

① Authentication & Users

Method	Endpoint	Description	Request Body / Params
POST	/api/auth/register/	Register new user	{ "userName", "userEmail", "userPhone", "userAddress", "password" }
POST	/api/auth/login/	Login user (JWT/Token)	{ "userEmail", "password" }

GE T	/api/users/me/	Get logged-in user profile	Auth required
PU T	/api/users/me/	Update profile	{ "userN ^u ame", "userPhone", "userAddress", "profilePicU RL" }
PO S T	/api/auth/password/forgot/	Request password reset	{ "userEmail" }
PO S T	/api/auth/password/reset/	Reset password using OTP/token	{ "token", "newPassword" }

PO /api/auth/otp/
S send/
T

Send OTP for verification

```
{ "userEmail" }
```

PO /api/auth/otp/
S verify/
T

Verify OTP

```
{ "userEmail", "otp" }
```

2 Roles & Pandits

Method	Endpoint	Description	Request Body / Params
GET	/api/pandits/	List all verified pandit s	Optional query: language, experienceYears

GE T	/api/pandits/<id>/	Get pandit detail s	-
PO S T	/api/pandits/	Create pandit profile	{ "experienceYears", "bio", "certificateURL", "languages": [] }
PU T	/api/pandits/<id>/	Update pandit profile	Same as POST
DE L E T E	/api/pandits/<id>/	Delete pandit profile	Admin only

③ Occasions, Samagri & Books

Occasions

Method	Endpoint	Description	Request Body / Params
GET	/api/occasions/	List all occasions	-
GET	/api/occasions/<id>/	Occasion detail + price	-
POST	/api/occasions/	Add occasion	{ "occasionName", "occasionDescription", "imageURL" }
PUT	/api/occasions/<id>/	Update occasion	Same as POST

DE	/api/occasions/<id>/	Delete occasion	Admin only
L		occasion	
E		ion	
T			
E			

Samagri & Books

Method	Endpoint	Description
GET	/api/samagri/	List all samagri (optional filter by occasion)
GET	/api/books/	List all books (optional filter by type)
GET	/api/samagri/<id>/	Samagri detail + price
GET	/api/books/<id>/	Book detail + price
POST/PUT/ DELETE	/api/samagri/ & /api/books/	Admin only, standard CRUD

4 Orders, Cart & Booking

Orders

Method	Endpoint	Description	Request Body / Params
POST	/api/orders/	Create new order (cart items + optional booking)	{ "userID", "items": [{ "itemType", "itemID", "quantity" }], "bookingID": optional }
GET	/api/orders/	List user orders	-
GET	/api/orders/<i>d</i>/	Order detail including bookings & cart items	-

Method	Endpoint	Description	Request Body / Params
PUT	/api/orders/<id>/	Update order status / payment confirmation	{ "status", "paymentStatus" }

Method	Endpoint	Description	Request Body / Params
DELETE	/api/orders/<id>/	Cancel order	-

Cart

Method	Endpoint	Description	Request Body / Params
POST	/api/cart/	Add item to cart	-

Method	Endpoint	Description	Request Body / Params
POST	/api/cart/	Add item to cart	{ "orderID", "itemType", "itemID", "quantity" }

Method	Endpoint	Description	Request Body / Params
GET	/api/cart/<orderID>/	Get cart	-

		detai	-
		ls	-
DE	/api/cart/<c artItemID> /	Remov e item	-
L			
E			
T			
E			

Booking

Met h o d	Endpoint	Description	Request Body / Params
PO S T	/api/booki ngs/	Create booking (pandit + occasion, optional orderID)	{ "userID", "panditID", "occasionID" , "datetime", "orderID": optional }
GE T	/api/booki ngs/	List user bookings	-

GET	/api/bookings/<id>/	Booking details (video room info included)	-
PUT	/api/bookings/<id>/	Update booking (reschedule/status)	{ "datetime", "status" }
DELETE	/api/bookings/<id>/	Cancel booking	-

5 Payments (Khalti / Stripe)

Method	Endpoint	Description	Request Body / Params
POST			

PO S T	/api/payments/	Make payment	{ "orderID", "gateway", "amount", "currency" }
GE T	/api/payments/ <id>/	Payment status	-
GE T	/api/orders/<i d>/payments/	Payments for an order	-

⑥ Video Rooms & Chat

VideoRoom

Me t h o d	Endpoint	Description	Request Body / Params
GE T	/api/videorooms /<bookingID>/	Room URL &	-

		participan ts
PO S T	/api/videorooms /	Create room for booking <pre>{ "bookingID", "panditID" }</pre>
PU T	/api/videorooms /<roomID>/	Update room / add recording URL <pre>{ "recordingURL", "endTime" }</pre>

Video Participants

Me t h o d	Endpoint	Description	Request Body / Params
PO S T	/api/videoparticip ants/	Add partici pant	<pre>{ "roomID", "userID" }</pre>

GE /api/videoparticipants/<roomID>/ List participants

ChatMessage

Method	Endpoint	Description	Request Body / Params
POST	/api/chat/	Send message	{ "bookingID", "senderID", "text", "imageURL" }

7 Review, Kundali & Alerts

Review

Method	Endpoint	Description	Request Body / Params
POST	/api/reviews/	Add review for booking	{ "bookingID", "rating", "comment" }
GET	/api/reviews/<bookingID>/	List reviews for a booking	-

Kundali

Method	Endpoint	Description	Request Body / Params
POST	/api/kundali/	Create Kundali	{ "dob", "time", "place", "userID" }

GE	/api/kundali	Get user	-
T	<userID>/	Kunda lis	

Alert

Me t h o d	Endpoint	Description	Request Body / Params
GE	/api/alerts/	List alerts	-
T	<panditID>	for	
	/	pandit	
PO	/api/alerts/	Create alert	{ "panditID",
S		(admin/p	"type",
T		andit)	"message",
			"language" }

Notes for Exam

1. Follow **REST principles**: resources = nouns, HTTP verbs for actions.

2. Use **JWT token authentication**.
3. Support **pagination & filtering** on large lists (Pandits, Orders).
4. **Booking optionally linked to orderID** → allows buying items only.
5. **Payments linked to orderID** → unified checkout for books, samagri, and pandit booking.