Backend Project API Documentation

Setup & Installation

1. Clone the repository or copy project files.
2. Run `npm install` to install dependencies.
3. Create a `.env` file in project root with the following keys:
   - MONGO_URI: Your MongoDB connection string
   - JWT_SECRET: Secret key for JWT
   - JWT_EXPIRE: JWT token expiration (e.g., '7d')
   - CLIENT_URI: Your frontend URL (for CORS)
   - STRIPE_SECRET_KEY: Your Stripe secret API key
   - STRIPE_WEBHOOK_SECRET: Your Stripe webhook signing secret
4. Run the server with `node server.js` or `nodemon server.js`.

Make sure MongoDB is accessible and `.env` variables are correct.

API Endpoints

Authentication

POST /api/auth/register

- Body: { name, email, password }
- Registers a new user.

POST /api/auth/login

- Body: { email, password }
- Logs in a user and sets JWT token cookie.

POST /api/auth/logout

- Logs out user by clearing token cookie.

Courses

GET /api/courses

- Public: List all courses.

GET /api/courses/:id

- Public: Get course details by ID.

POST /api/courses

- Protected (admin/instructor)
- Body: { title, description, price }

- Create new course.

PUT /api/courses/:id

- Protected (admin/instructor)
- Update course info.

DELETE /api/courses/:id

- Protected (admin/instructor)
- Delete course.

Orders (Purchases)

POST /api/orders

- Protected
- Body: { courseId, paymentIntentId, amount }
- Create new order after payment.

GET /api/orders/myorders

- Protected
- Get current user's order history.

Payments (Stripe)

POST /api/payments/create-checkout-session

- Protected
- Body: { courseId, price, success_url, cancel_url }
- Creates a Stripe checkout session.

POST /api/payments/webhook

- Public
- Stripe calls this webhook to confirm payment success.

Authentication & Testing Notes

- All protected routes require the JWT token cookie set by login.
- Use Postman or similar to test endpoints.
- For Stripe webhook, use Stripe CLI or expose webhook publicly.
- To test payment flow, first create a course, then create a checkout session.
- After payment success, orders are created automatically via webhook.
- Use Postman environment to store cookies for authenticated requests.

Running the Server

Run:

```
node server.js
```

or

```
nodemon server.js
```

Make sure MongoDB and environment variables are correctly configured.

Access API via http://localhost:5000 or your configured port.