**Secure login for users and admins using JWT**

**Current State:**

* Your backend (server.js) only has GitHub OAuth for authentication, and it does not use JWT for session management.
* No user/admin role management or JWT middleware is present.

**Missing/Needs Change:**

* Implement user and admin authentication (not just GitHub OAuth).
* Generate and return JWT tokens upon successful login.
* Store JWT in localStorage/cookies on the frontend.
* Add middleware to verify JWT and enforce role-based access (e.g., admin panel).

**2. Upload Excel files (.xls, .xlsx) using Multer**

**Current State:**

* No endpoints or middleware for file uploads are present.
* No Multer integration.

**Missing/Needs Change:**

* Add an endpoint for file uploads using Multer middleware.
* Configure Multer to accept only .xls and .xlsx files.

**3. Parse and read data using SheetJS**

**Current State:**

* No code for parsing Excel files or using SheetJS (xlsx).

**Missing/Needs Change:**

* After upload, use SheetJS to parse the Excel file on the backend.
* Convert the parsed data to JSON for frontend use.

**4. Let users select which columns to visualize (X and Y axes)**

**Current State:**

* The frontend has components for dashboards and chart selection, but it’s unclear if dynamic column selection is implemented.

**Missing/Needs Change:**

* Ensure that after parsing, the frontend displays dropdowns for all column headers.
* Allow users to select X and Y axes for chart generation.

**5. Generate interactive 2D/3D charts using Chart.js and Three.js**

**Current State:**

* The frontend has chart-related components, but it’s unclear if both Chart.js (2D) and Three.js (3D) are integrated.

**Missing/Needs Change:**

* Integrate Chart.js for 2D charts and Three.js for 3D charts.
* Feed parsed data to these libraries based on user selection.

**6. Allow users to download charts as PNG or PDF**

**Current State:**

* No evidence of chart export functionality.

**Missing/Needs Change:**

* Implement chart export as PNG using Chart.js methods.
* Add PDF export using jsPDF or html2canvas.

**7. Maintain upload and analysis history for each user**

**Current State:**

* No backend logic for storing upload or analysis history.
* No MongoDB integration for metadata storage.

**Missing/Needs Change:**

* Store metadata (file name, upload date, selected axes, chart type) in MongoDB.
* Display this history on the user’s dashboard.

**8. Admin dashboard to view usage statistics and manage users**

**Current State:**

* No admin dashboard or user management endpoints/components.
* No statistics or CRUD operations for users.

**Missing/Needs Change:**

* Create an admin dashboard view.
* Implement endpoints to fetch stats (uploads, active users, chart types).
* Add CRUD operations for user management (block/delete).

**Summary Table**

| **Requirement** | **Status** | **What’s Missing/Needs Change** |
| --- | --- | --- |
| JWT Auth & Roles | ❌ | Add JWT, user/admin roles, middleware |
| Excel Upload (Multer) | ❌ | Add Multer, upload endpoint |
| Excel Parsing (SheetJS) | ❌ | Add SheetJS parsing logic |
| Column Selection | ❓ | Ensure dynamic dropdowns for columns |
| 2D/3D Charts | ❓ | Integrate Chart.js & Three.js |
| Chart Download | ❌ | Add PNG/PDF export functionality |
| Upload/Analysis History | ❌ | Store metadata in MongoDB, show in dashboard |
| Admin Dashboard | ❌ | Create admin view, stats, user management |

**Legend:**

* ❌ = Missing
* ❓ = Unclear/Needs Verification