Absolutely. Here’s a **detailed, production-ready Replit Agent prompt** tailored to the **exact fixes needed**, based on the conversation log and current system errors across the **Chargebee, MySQL saved queries, annotations, WebSocket, and root health check systems**.

**✅**

**Replit Agent Prompt — Fix Critical Integration Bugs for Recurrer Platform**

**🧩 Project:** Recurrer – Customer Success OS

**📍Objective:** Fix integration bugs across WebSocket annotations, Chargebee API, MySQL Saved Queries, Health Check Endpoint, and improve error handling and validation in backend and frontend.

**🛠️ Areas to Fix**

**🔹 1.**

**Fix Annotation Reply Creation Error**

**Issue**

POST /api/annotations/3/replies 400 :: {"error":"Invalid annotation data"}

**Fix**

* **File:** server/annotations.ts
* In createAnnotationReply handler:
  + Ensure entity\_type and entity\_id are **not expected from the request body**.
  + Instead, fetch from **parent annotation**.
  + Use Zod .passthrough() on schema to ignore extra client fields.
* In client:
  + **File:** client/src/components/annotations/annotation-item.tsx
  + Remove entity\_type and entity\_id from payload in handleReplySubmit().

**🔹 2.**

**Fix Chargebee API Integration Failing on Missing Credentials**

**Issues**

* Environment variables missing: CHARGEBEE\_API\_KEY, CHARGEBEE\_SITE
* Failing with:

Error calling Chargebee API: 404 Not Found

**Fix**

* **File:** server/chargebee.ts
  + Add validation: if API key or site is missing, throw friendly error.
  + Improve error logging on makeRequest method: return response body if 4xx/5xx error.
* Add CHARGEBEE\_API\_KEY and CHARGEBEE\_SITE to .env via Replit Secrets.
* Improve /api/admin/chargebee-config/test:
  + Instantiate new ChargebeeService with provided credentials in request body instead of global singleton.

**🔹 3.**

**Fix MySQL Saved Query “Unexpected Token” HTML Error**

**Issue**

Unexpected token '<', "<!DOCTYPE "... is not valid JSON

Likely cause: backend is returning an HTML error page, not a JSON response (e.g. due to invalid endpoint, auth redirect, or Vite handling frontend routes).

**Fix**

* **File:** server/routes.ts
  + Verify /api/admin/mysql-saved-queries/:id/run returns proper JSON.
  + Wrap query execution in try/catch and handle malformed SQL or bad params.
  + Ensure proper headers:

res.setHeader('Content-Type', 'application/json');

* **Client:** client/src/pages/admin.tsx
  + Ensure saved query results are parsed only when response is application/json.
  + Add try/catch to prevent breaking UI when JSON fails.

**🔹 4.**

**Fix Health Check Endpoint Conflict**

**Issue**

Frontend not loading, health check hijacks / route.

**Fix**

* **File:** server/routes.ts
  + Replace:

app.get('/', ...) // REMOVE

* + With:

app.get('/api/health', (\_, res) => res.json({ status: 'ok', message: 'Recurrer API is running' }));

* Ensure Vite middleware is properly serving frontend after API routes.

**🔹 5.**

**Fix Chargebee and MySQL Table Migration Errors**

**Issue**

error: relation "chargebee\_config" already exists

**Fix**

* **Files:**
  + server/migrate-chargebee-config.ts
  + server/migrate-mysql-config.ts
* Modify table creation:

try {

await pool.query(`CREATE TABLE IF NOT EXISTS chargebee\_config (...)`);

} catch (err) {

if (err.code !== '42P07') {

console.error('Failed to create table:', err);

}

}

**🔹 6.**

**Fix WebSocket Initialization Error in Node.js**

**Issue**

ReferenceError: WebSocket is not defined

**Fix**

* **File:** server/routes.ts
* At the top:

import WebSocket, { WebSocketServer } from 'ws';

* Replace all direct usage of WebSocket (which is browser-only) with the imported class.

**🔹 7.**

**Improve Annotation WebSocket Integration**

**Fix**

* Create a shared WebSocket context in React:
  + client/src/providers/AnnotationWebSocketProvider.tsx
* Ensure StandaloneAnnotationList uses the WebSocket context.
* On disconnection, show message and retry after 3 seconds.

**🔹 8.**

**Fix Saved Query Auto-select in Admin Panel**

**Fix**

* **File:** client/src/pages/admin.tsx
* Add null checks before accessing query fields:

if (selectedQuery && selectedQuery.query) {

setEditorValue(selectedQuery.query);

}

* Add fallback message if no query is loaded:

{!selectedQuery && <p>No query loaded.</p>}

**🔹 9.**

**Improve Logging & Debugging**

* Add logging in:
  + Annotation reply creation
  + Chargebee API test endpoint
  + Saved query execution
* Example:

console.log('Running saved query:', query);

console.log('Chargebee config:', config);

**✅ Confirmed Working Features (Do Not Overwrite)**

* Health check endpoint moved to /api/health
* WebSocket connection live for annotations
* Frontend dashboard accessible
* Chargebee credentials now set via Replit Secrets

**🧪 Final Testing Checklist**

* Annotations can be added and replied to
* Chargebee config can be saved and tested
* MySQL saved queries execute and show result without crashing
* App loads frontend on root URL /
* Migrations don’t crash when re-run
* Health check works at /api/health

**🎯 Expected Output**

* Backend code in server/routes.ts, chargebee.ts, annotations.ts, storage.ts
* Client-side code in admin.tsx, annotation-list.tsx, annotation-item.tsx
* WebSocket setup fixed and tested
* App is stable and error-free on startup and interactions

Let me know if you’d like this prompt exported as a .md doc, or scaffolded as a ticket backlog for phased delivery.