CreditPathAI – Actions, Logging, and Error Handling Report

1. Actions

- **Purpose:** Predict credit default risk for borrowers based on input data.
- **Frontend Actions:**
    - Collect borrower details via a form (including `loan_id`).
    - Provide default values for quick input.
    - Dropdowns for categorical columns to ensure valid selections.
    - Validate all fields are filled; highlight missing fields in red.
    - Submit data to the backend for prediction.
- **Backend Actions:**
    - Accept multiple borrowers at once (`POST /predict`).
    - Preprocess data:
        - Clean `emp_length` and `term`.
        - Drop irrelevant columns (`address`, `emp_title`, `title`).
        - Scale numeric features using the pre-trained scaler.
        - One-hot encode categorical features aligned with the training model.
    - Predict default probability using a trained LightGBM model.
    - Map probability to **risk level** and **recommended action**:
        - Low risk → "Send reminder"
        - Medium risk → "Call borrower"
        - High risk → "Prioritize collection / restructure loan"
- **Frontend Display:**
    - Show results for each borrower:
        - `loan_id`, default probability, risk level, recommended action.

---

2. Logging

- **Purpose:** Keep a persistent record of predictions for auditing and monitoring.
- **Log Content:** Only essential information is saved in `predictions.log`:
    - `loan_id`
    - Borrower input data

    - Predicted default probability
    - Risk level
    - Recommended action
- **Format Example:**

```
2025-10-12 10:57:54 - {'loan_id': 'LN123', 'borrower': {...},
'default_probability': 0.169, 'risk_level': 'Low risk', 'recommended_action':
'Send reminder'}
```

---

3. Error Handling

- **Frontend:**
    - Highlights missing or empty fields in red.
    - Shows a user-friendly message: “Please fill in all required fields.”
    - Handles failed backend requests gracefully with: “Failed to fetch. Please try again.”
- **Backend:**
    - **Validation Errors:**
        - If any required field is missing or invalid, a `400 Bad Request` is returned with a descriptive message.
    - **Unexpected Errors:**
        - Logged with full traceback in the backend log (not in `predictions.log`).
        - Returns a generic error message to the frontend: “An unexpected error occurred.”
- **Exception Logging:**
    - Only validation or unexpected errors are logged with `logging.warning` or `logging.error`.
    - Prevents frontend crashes or unhandled exceptions.

---

4. Summary

- End-to-end pipeline is fully functional:
    1. **Data input → Validation → Preprocessing → Prediction → Result display → Logging.**
- Logging is **structured, clean, and unique per borrower** (`loan_id`).
- Error handling ensures **robust frontend experience** and **safe backend operation**.
- System is ready for **analytics, reporting, or deployment**.