

Fraud Lens — End-User Manual

Sujay S

May 14, 2025

Contents

1	Introduction	2
2	System Requirements	2
3	Accessing the Application	2
4	Creating an Account	2
5	Making a Fraud Prediction	4
6	Giving Feedback	4
7	Understanding the Prediction (Explain Tab)	5
8	Monitoring Metrics	5
9	Signing Out	6
10	Troubleshooting	6
11	FAQ	7

1 Introduction

Fraud Lens is a web-based application that helps you identify potential credit-card fraud, understand why the model made its decision, and monitor overall system health. This manual walks you through the day-to-day tasks you can perform as an end user.

2 System Requirements

- Modern web browser (Chrome, Edge, Firefox, Safari) with JavaScript enabled.
- Access to the application URL: <http://localhost:3002> (or the address provided by your administrator).
- A valid e-mail address to complete OTP verification.

3 Accessing the Application

Open your browser and navigate to <http://localhost:3002>. You will land on the **Sign in** page (Figure 1).

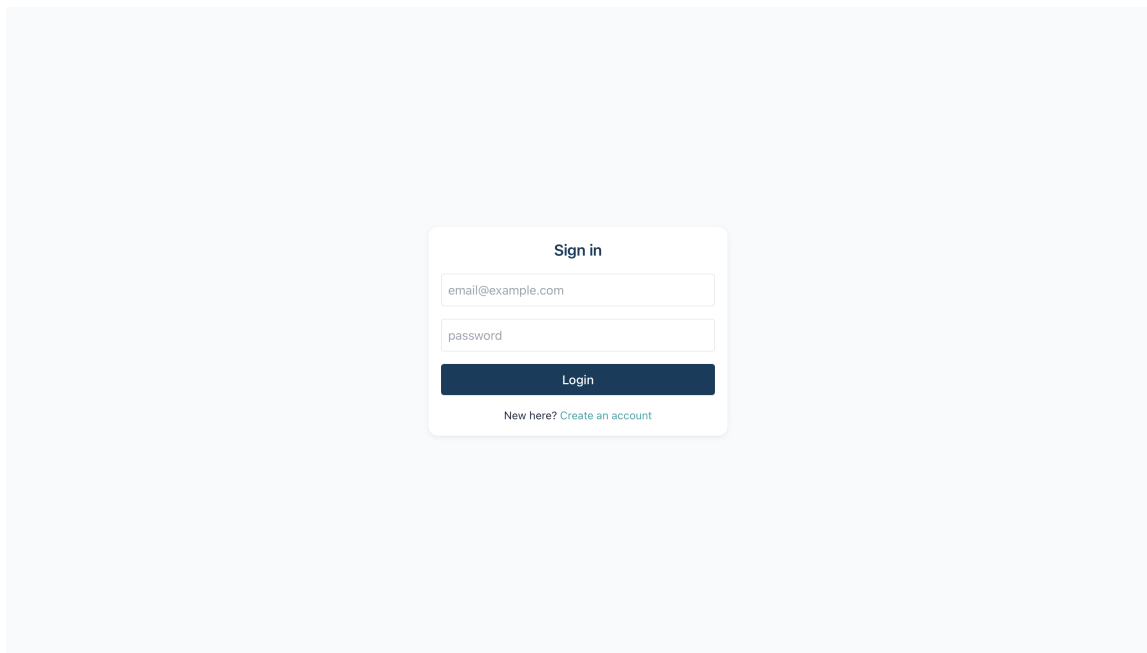


Figure 1: Sign-in page

If you already have an account, enter your credentials and click **Login**. Otherwise follow Section 4 to create an account.

4 Creating an Account

Click the *Create an account* link below the login form. Fill in the registration form (Figure 3) as follows:

1. **Name, Age, Gender, Country.**

2. Enter your e-mail and click **Send** to receive a 6-digit OTP. Check your inbox (and spam folder). Type the code when prompted.

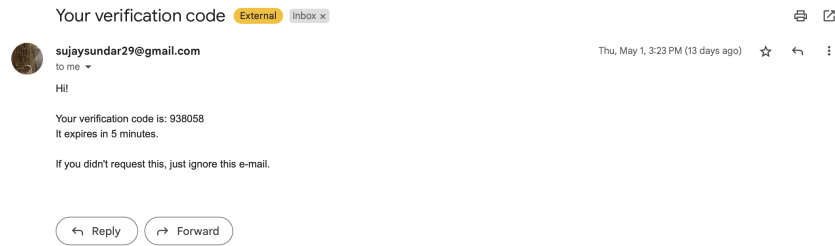


Figure 2: Email verification

3. Choose a strong password. The strength bar turns green when your password meets the minimum complexity.
4. Click **Register**. A welcome e-mail will confirm that your account is active.

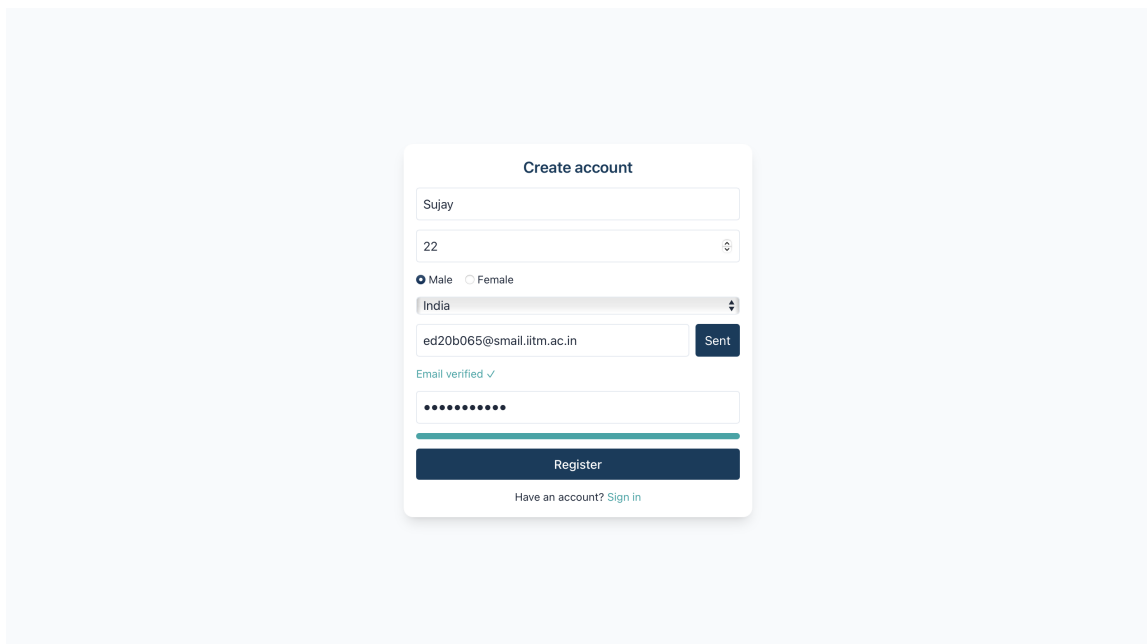


Figure 3: Registration form



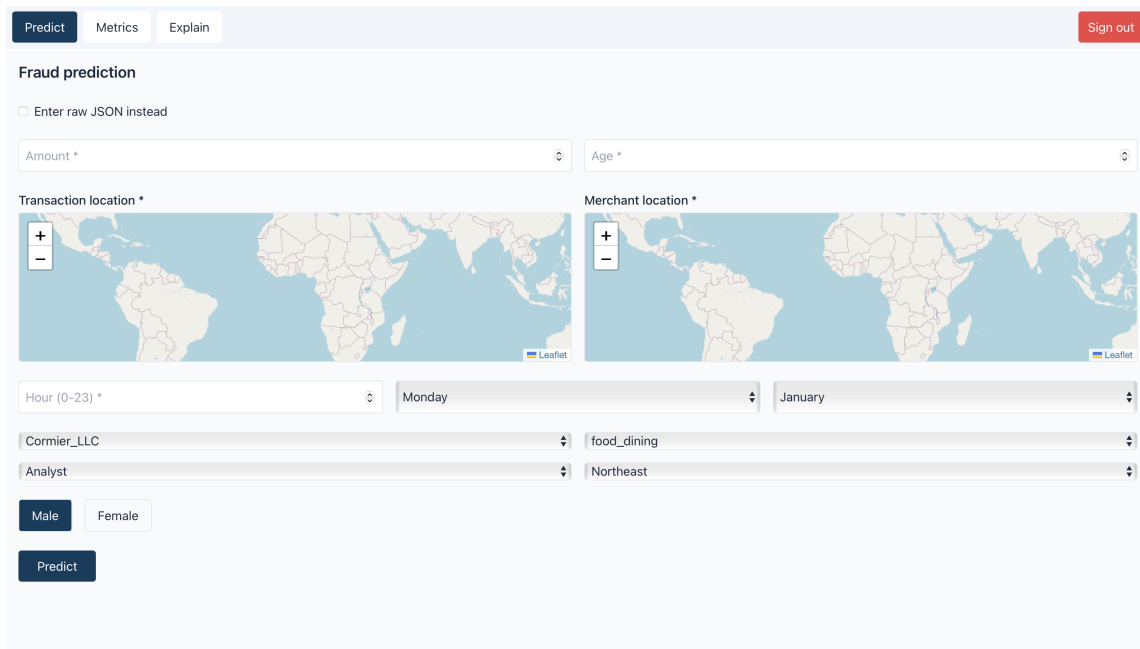
Figure 4: Welcome mail

After successful registration you are automatically redirected to the Sign-in page.

5 Making a Fraud Prediction

Once logged in you will see the *Predict* page (Figure 5). Provide the transaction details:

1. **Amount** in the currency of the transaction.
2. **Transaction location** and **Merchant location**: click on the map to drop a pin; coordinates are captured automatically.
3. **Age** of the cardholder, **Hour**, **Day of week**, **Month**.
4. Select **Merchant**, **Category**, **Job**, and **Region** from the drop-down lists.
5. Select cardholder **Gender**.
6. Click **Predict**. The result shows the fraud probability and the predicted label (**fraud** or **not_fraud**).

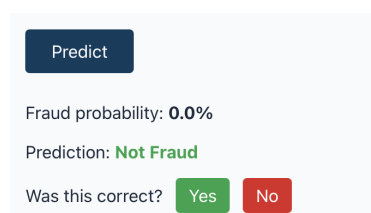


The screenshot shows the 'Predict' page of a web application. At the top, there are three tabs: 'Predict' (active), 'Metrics', and 'Explain'. A 'Sign out' button is in the top right corner. Below the tabs, the page is titled 'Fraud prediction'. There is a checkbox labeled 'Enter raw JSON instead'. The form contains several input fields and maps: 'Amount *' and 'Age *' are text inputs; 'Transaction location *' and 'Merchant location *' are maps with a pin icon; 'Hour (0-23) *' is a text input; 'Monday', 'January', 'Cormier_LLC', 'food_dining', 'Analyst', and 'Northeast' are dropdown menus; 'Male' and 'Female' are radio buttons; and a 'Predict' button is at the bottom.

Figure 5: Predict page

6 Giving Feedback

Below the prediction a short feedback form asks whether the model was correct. Select **Yes** or **No**. Your feedback improves the model metrics shown on the *Metrics* page.



The screenshot shows a feedback form. It has a 'Predict' button at the top. Below it, the text 'Fraud probability: 0.0%' is displayed. Then, 'Prediction: Not Fraud' is shown, with 'Not' in green and 'Fraud' in red. At the bottom, the text 'Was this correct?' is followed by two buttons: 'Yes' (green) and 'No' (red).

Figure 6: Feedback form

7 Understanding the Prediction (Explain Tab)

Click the **Explain** tab. The application auto-fills a natural-language prompt containing the transaction features and model output. Press **Explain**. Within a few seconds Gemini 2.0 Flash returns a plain-English justification summarising the most influential factors.

Predict Metrics **Explain** Sign out

Explainable AI

You are an explainable-AI assistant for a credit-card fraud-detection model. Given the JSON representation of the transaction features and the model's output, explain—at a business-analyst level—*why* the model predicted it as **FRAUD** (probability 1.00). Focus on the most influential features and avoid deep math jargon.

Transaction JSON:

```
```json
{
 "amt": 1234.0,
 "tx_hour": 1,
 "merchant_category": "food_dining",
 "customer_age": 22,
 "customer_gender": "Male",
 "customer_country": "India",
 "merchant_lat": 41.51,
 "merchant_long": -75.94,
 "customer_lat": 35.24,
 "customer_long": -90.35
}
```

**Explain**

Okay, based on the provided transaction data, here's why the fraud detection model flagged this transaction as fraudulent with a probability of 1.00:

**Key Indicators of Fraud:**

- Large Transaction Amount:** The transaction amount is \$1234.0. This is considered a large amount compared to typical spending patterns, especially for someone relatively young (22 years old). Larger transactions are often a red flag for fraud.
- High Displacement:** The distance between the customer's location (lat: 35.24, long: -90.35) and the merchant's location (lat: 41.51, long: -75.94) is quite substantial. This suggests the customer and merchant are in drastically different geographic locations. Fraudsters often use stolen card details to make purchases far from the cardholder's usual location.
- Transaction Hour:** The transaction occurred at 1 AM (tx\_hour = 1). Transactions occurring at unusual hours, like late at night, are more likely to be fraudulent.
- Merchant Category:** The transaction is categorized as "food\_dining". Fraudsters often test stolen cards with smaller purchases in categories like food and dining before attempting larger, more noticeable transactions. This feature alone is not strong, but in combination with other factors, it adds to the suspicion.
- Job type: Analyst:** Fraudsters are more likely to use younger people's cards that have less fraud detection on them due to them having less financial history.

**In Summary:**  
The model likely identified this transaction as fraudulent because of the combination of a **large transaction amount**, a **significant distance between the cardholder and the merchant**, an **unusual transaction time**, the fact that the customer is young, and the **merchant category**. Individually, some of these factors might not be enough to trigger a fraud alert, but together they paint a picture of a high-risk transaction.

Figure 7: Explain tab

## 8 Monitoring Metrics

Open the **Metrics** tab to view live dashboards embedded from Grafana:

- **User details:** email and sign-in status.
- **API usage:** Predict and Explain call counts.
- **Model metrics:** accuracy, F1-score, true/false positives and negatives.
- **System health:** CPU load, memory, network I/O, disk activity scraped from Node Exporter.

Panels auto-refresh every 5 seconds. Hover over any graph to inspect exact values.

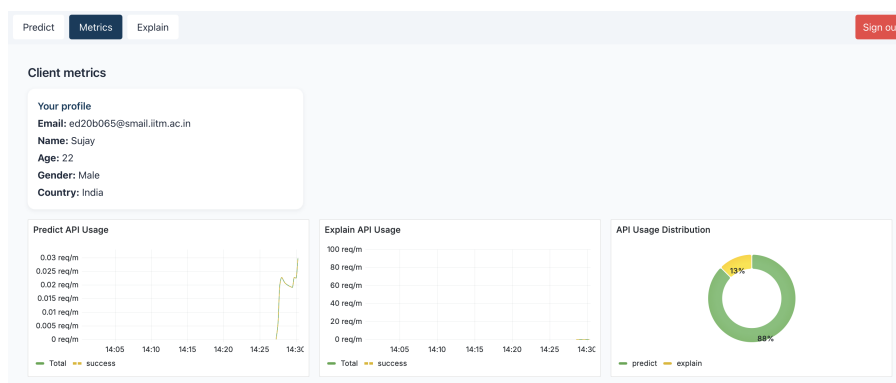


Figure 8: Client metrics

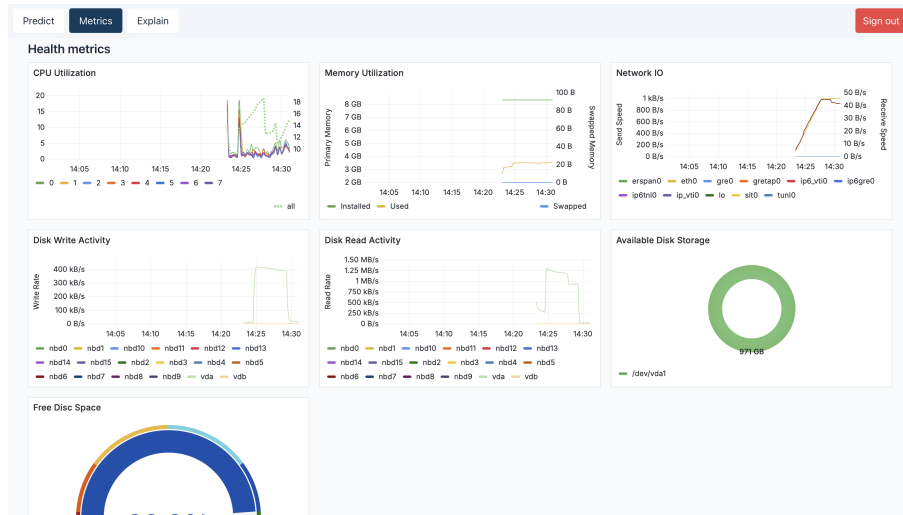


Figure 9: Health metrics

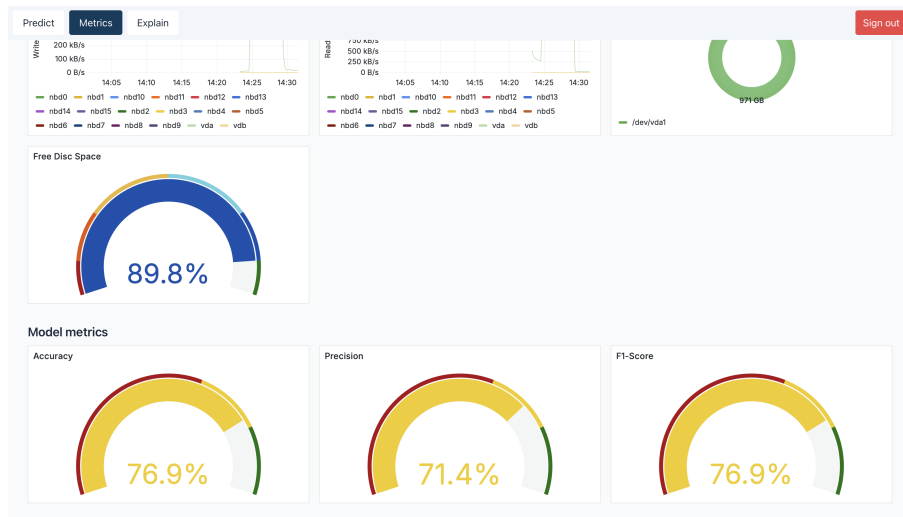


Figure 10: Model metrics

## 9 Signing Out

Click the red **Sign out** button in the top-right corner. You will be redirected to the Sign-in page and your JWT session token will be cleared.

## 10 Troubleshooting

**Didn't receive OTP** Wait a minute and check your spam folder. Use the *Resend* button; you may request up to 5 codes per hour.

**401 Invalid credentials** Double-check your e-mail and password. If you've forgotten your password, contact the administrator—password reset isn't yet automated.

**500 Gemini error** The explanation service may be temporarily unavailable. Retry after a few minutes.

**Maps not loading** Make sure your browser allows third-party map tiles (e.g. OpenStreetMap).

**Ports not available** The service failed to bind to its port. Identify the occupying process with:

```
sudo lsof -i :<port>
```

Then terminate it (e.g. `kill <PID>`) and restart the service.

**Database initialization failed** The database container may not have started correctly. Re-build and restart all services with:

```
docker-compose up --build
```

Check the logs for specific errors and ensure volumes are mounted with correct permissions.

## 11 FAQ

- **Is my password stored securely?** Yes, passwords are hashed with bcrypt; the plain text is never stored.
- **Why do I need to give feedback?** Feedback helps measure real-world performance and could trigger model retraining.
- **Can I bulk-upload transactions?** Not yet; this version is designed for single-transaction analysis.

## Enjoy using Fraud Lens!

If you encounter issues not covered here, file a ticket on the project's GitHub or e-mail the support team at [sujaysundar29@gmail.com](mailto:sujaysundar29@gmail.com).