Ethical Issues

Privacy

The GURU system processes sensitive user and business data, including sales records, inventory, and customer information. Because it integrates third-party APIs such as e-commerce and AI platforms, privacy is a primary ethical concern. Users expect that their business and customer data will be kept secure, used only for stated purposes, and not retained indefinitely.

Solutions:

- Reasonably collect only essential data for analytics and insights
- Encryption in transit and at rest to protect stored business data.
- Provide transparent privacy disclosures explaining data collection, storage, and retention.
- Allow users to request their data deletion or exportation

Fairness and Non-Discrimination

The GURU AI uses data-driven predictions and recommendations to help businesses improve performance. However, algorithmic outputs can unintentionally favor certain demographics, product types, or market segments, leading to bias or unfair outcomes.

Solutions:

- Conduct bias and fairness audits on AI models and datasets.
- Include diverse and representative data during model training.
- Provide explainable AI outputs so users can further reason how recommendations are generated.
- Require human review for automated actions such as pricing or promotions

Misuse and Public Harm

Automation and AI features could be misused to harm others by sending unsolicited promotional emails, leaking sensitive data, or over-relying on inaccurate forecasts.

Solutions:

- User confirmation for automated communications and data exports.
- Applied rate limits, role-based access control, and audit logs to prevent abuse.
- Accuracy indicators and disclaimers with AI predictions to avoid over-reliance.

Legal Issues

Licensing and Intellectual Property

Third-party libraries and frameworks used in our app have their respective licenses.
We have to examine each license in order to ensure that we can use, modify, and distribute them.

For open-source licensing such as MIT and Apache 2.0 licensing, we can use the licensed materials free of charge as long as attribution requirements are met. Datasets as well as APIs used must also be reviewed for usage rights. We will maintain a LICENSES.md file containing the list of all our dependencies and their corresponding licenses.

Privacy and Data Protection Laws

Our app will have to adhere to the world's prominent laws concerning user privacy, and in this case it’s CCPA for users in California. Such regulations compel developers to be open about our use of personal information. We will give the user a Privacy Policy about their information collection, storage, and deletion.

User Content and Legal Responsibility

Before accessing our app, the user must agree to our “Terms of Use”. They must only upload content that they own or have the right to use. In order to protect our service, we will comply with the DMCA guidelines for copyright complaints.

Security Issues

Securing, and encrypting our user’s sensitive information is a key goal in our design philosophy. Unwanted access to this information can jeopardize business security and reveal sensitive information about businesses.

Information the app will store includes customer information, such as names and emails, login credentials, and intricate business analytics. Other protections at the development level can include hiding sensitive information by using environment variables and not hardcoding.

We must also be sure the .env file is in .gitignore when committing code to github for security reasons, and that user input must also be sanitized to prevent SQL injection and cross-site scripting. In addition to sanitizing input, we could look to store hashed passwords and not plain text for extra security encryption. As well as use HTTPS to encrypt communication between the client and server.

Rate limiting is a common security measure to limit log in attempts to prevent brute force attacks on the app. This will also be important to use sessions or JSON Web Tokens (JWTs) to keep track of logged-in users

Prevention

Considerations for the app that must taken into account for the security of the user's data and the app in general.

Attack vectors that an advisory might try to use include:

- The login page to access an account that might not belong to them
- The application itself through the use of AI agent, unprompted request to break the system
- AI agent will provide inaccurate information to the user

The biggest concern that is identified is prompt injection. This concern is both to a user who inadvertently breaks the system or a hacker who does it for malicious purposes. This can include making a request that ignores previous rules set for the AI and will execute regardless.
This is unique to LLM because we prompts are not generated the same. Output is probabilistic and getting the same exact response does not happen. A user can input anything.

This is similar to another security risk to account for is improper output handling. Without validating or sanitizing outputs then this will become a user can retrieve information through the LLM.

Another possibility is misinformation. LLM are not perfect and could give wrong/ill-advised answers and present it as fact. This spells problematic for the user as they wouldn't know better without verifying the information themselves, however it spells even more legal trouble.

Security Framework in Use :
https://genai.owasp.org/llmrisk/llm01-prompt-injection/
https://genai.owasp.org/llmrisk/llm052025-improper-output-handling/
https://www.stackhawk.com/blog/top-llm-security-risks/

Using the OWasp LLM top ten along with the guidelines to resolve the issues as follows:

Login Page:

- Encryption through OpenSSL
- OAuth for authentication or GoogleAuth
- Input validation DOMPurify for React validator.js

- Password Strength I

AI agent Prompt injection

- Require human approval for high risk actions - credential proof
- Implementation of input and output filters, immutable instructions
- Utilization of OpenAI moderation API

Improper Output Handling

- Apply the concept of zero trust by applying input validation
- Restrict model to certain callable functions that it can access and execute