GRNLITE

Revolutionizing the Feedback Process for Independent Authors

Team Members:

Eric Freeland - Frontend Developer
James Sanders - Backend Developer
Kelsea Conrad - Backend Developer

THE CHALLENGES FOR INDEPENDENT AUTHORS

- **Gatekeeping in Publishing:** Traditional publishing houses often impose barriers that prevent new authors from breaking into the market, leaving them to navigate the journey alone.

- **Finding Feedback:** Struggling to locate qualified beta readers and editors who align with their needs.

- **Managing Revisions:** Tracking feedback, consolidating insights, and applying revisions is time-consuming and overwhelming.

- **Burnout Risk:** Many authors face frustration and missed opportunities, leading many to abandon their manuscripts before publishing.

EMPOWERING AUTHORS

Find Your Perfect Match

- Connect with professionals tailored to your genre and needs.

Streamline Collaboration

- Securely share manuscripts and manage communication seamlessly.

Polish to Perfection

- Transform manuscripts into masterpieces with guided feedback and revision tools.

OUR GOALS

Networking with Professionals

Provide authors with a centralized platform to network with beta readers.

Quality Feedback

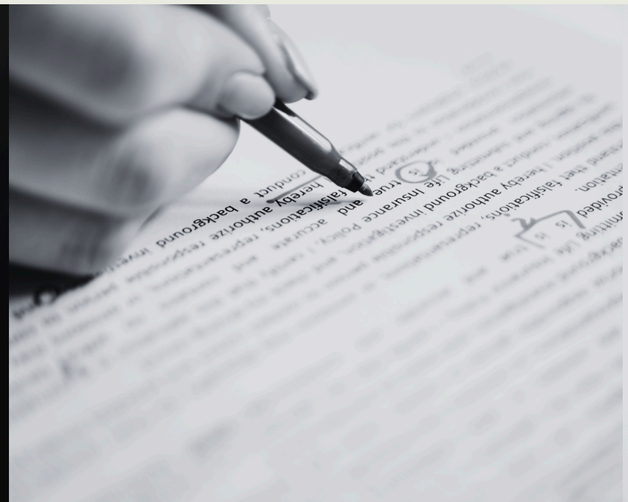
Ensure quality feedback through guided forms for beta readers.

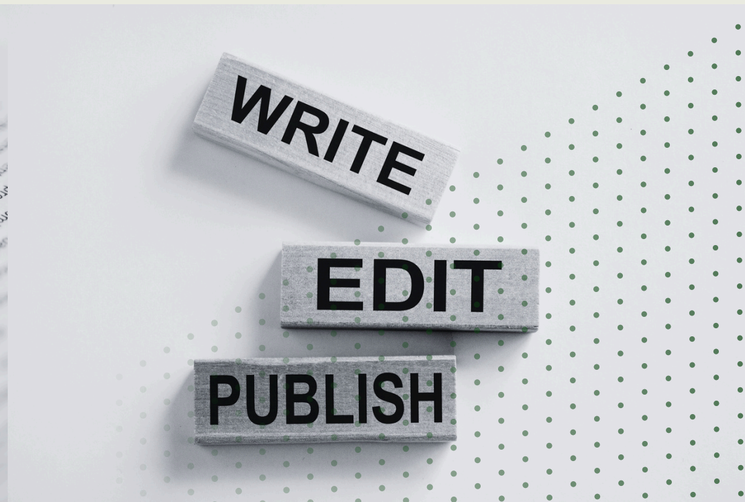
Improving Efficiency

Reduce the time and effort needed to organize and revise drafts by consolidating all feedback in one place.

More Great Stories in the World

Help bring more captivating stories to the world by empowering authors to turn their creative visions into polished masterpieces.





WHY WE CHOSE THIS PROJECT

Passion for Storytelling

- This project stems from a deep passion for storytelling and supporting authors as they bring their creative visions to life.
- As someone already immersed in this niche, I’ve seen firsthand the challenges authors face in refining their manuscripts and managing the feedback process.

Identified a Clear Need

- Independent authors often lack the resources, structure, and support needed to polish their drafts.
- The existing solutions fail to provide a seamless, secure, and tailored experience for both authors and beta readers.

WHY WE CHOSE THIS PROJECT

A Unique Opportunity

- GRNLITE bridges the gap between authors and beta readers, filling a niche that can elevate the quality of stories while empowering creators to share their work with the world.

APPLICATION DEMO

FRONTEND TECH STACK

WHY JAVASCRIPT?

Template Folder Structure

Reason for Choosing This Structure:
This folder structure prioritizes clarity, collaboration, and scalability. It ensures teams can easily navigate updates, maintain consistent titling, and provide backend developers with clean, endpoint-aligned HTML files for seamless integration.

Key Features

- **Backend Developer Friendly:**
  - Each HTML file is treated as an individual webpage, directly linked to a unique backend endpoint.
  - Simplifies redirection and backend-to-frontend communication.
- **Static Folder Functionality:**
  - Houses CSS, JS, images, and fonts for central access.
  - Enhances the modularity of design and functionality.

BACKEND TECH STACK

WHY DJANGO?

Efficient Development:

Django is a high-level Python framework that streamlined GRNLITE's backend development while ensuring scalability and security.

Built-in Features:

- **Django ORM:** Simplified database operations, making it easier to interact with PostgreSQL.
- **Authentication:** Provided robust user authentication and authorization tools.
- **Admin Panel:** Accelerated project management and testing during development.

WHY DJANGO?

A Learning Challenge:

We chose Django not just for its features but also to challenge ourselves to learn a new and widely-used framework. This project offered an opportunity to expand our technical skills and explore a tool none of us had used before.

WHY POSTGRESQL?

Relational Database:

PostgreSQL was ideal for managing GRNLITE's structured data, including:

- **Users:** Authors and beta readers.
- **Manuscripts:** Metadata, files, and feedback.
- **Feedback:** Detailed responses tied to specific projects and beta readers.

Advanced Features:

- Support for complex queries, making it easier to filter feedback and match beta readers to manuscripts.
- Scalability for handling future growth as GRNLITE expands.

FRONTEND DEVELOPER:

ERIC FREELAND

ROLE

My role to the project was to do the front end for the app. I used front end technologies such as HTML, CSS, and Javascript. HTML for the main structure of the app, CSS to style the app and flush out the over all look of the app, and Javascript for the functions of the app on the front end part.

WHAT I LEARNED

Building the frontend of an app using HTML, CSS, and JavaScript taught me how to structure content with semantic markup, create responsive and visually appealing layouts using CSS, and add interactivity through JavaScript. I learned to use DOM manipulation, event handling, and API integration to fetch and display dynamic data. By adopting modular practices, debugging with browser tools, and leveraging Git for version control, I ensured clean, scalable, and user-friendly development. This experience highlighted the importance of accessibility, responsiveness, and seamless collaboration between frontend and backend.

CHALLENGES

Building the frontend of an app comes with challenges like ensuring responsive design across devices, achieving cross-browser compatibility, and managing complex state dynamically. Debugging interactivity, optimizing performance, and handling API integration errors can be time-consuming. Maintaining clean, modular code while balancing fast development is crucial, especially on larger projects. Additionally, creating an intuitive, accessible user experience requires careful planning and iterations. Collaboration with a team may also bring version control conflicts, and staying updated with evolving tools and practices can be a continuous learning curve.

ACHIEVEMENTS

Achievements from building the frontend include creating a responsive, cross-browser interface, implementing dynamic features with JavaScript, and writing clean, reusable code. Optimizing performance and debugging challenges highlighted problem-solving skills. Successfully collaborating using Git and delivering a user-friendly frontend demonstrates strong development practices and web development proficiency.

IMPROVEMENTS

Some areas for improvement include enhancing performance by optimizing assets like images and scripts, improving code readability and organization for easier maintenance, and implementing more advanced features like state management or custom hooks. Additionally, refining the responsiveness for edge cases and testing on a wider range of devices and browsers would improve the user experience. Continuous learning and staying up-to-date with the latest tools and best practices in frontend development could also lead to more efficient workflows and better results.

BACKEND DEVELOPER:

JAMES SANDERS

ROLE

As a **Backend Developer**, I was responsible for:

- Maintaining GitHub repositories and managing operations such as deployments and integrations.
- Implementing beta reader functionality and focusing on authentication aspects using Django's built-in systems, ensuring seamless and secure user experiences.

WHAT I LEARNED

Django

- Working with Django has given me a deep appreciation for its versatility and the wealth of integrations it supports.
- Its comprehensive documentation has proven invaluable, allowing me to explore and implement diverse features effectively.

CHALLENGES

- One significant challenge was the disparity between local development and server-side deployments. Features that performed flawlessly in a local environment sometimes encountered issues during deployment due to differences in configurations and integration processes.
- The structure of an application plays a crucial role in its flow and usability—improper structuring can hinder performance and user experience, while thoughtful organization can enhance overall functionality.

ACHIEVEMENTS

- I gained a thorough understanding of Django's capabilities, limitations, and the deployment process.
- I honed my skills in user authentication, token handling, and security enhancements.
- Collaborating with a team taught me the importance of cohesive code planning and coordination, highlighting that every aspect of development—beyond the code itself—contributes to the success of a project.

IMPROVEMENTS

- Moving forward, I recognize the importance of detailed planning and structured collaboration.
- Establishing clear, team-approved expectations during the planning phase will be a priority in future projects to ensure alignment and smooth execution.

BACKEND DEVELOPER:

KELSEA CONRAD

ROLE

Primary Focus Areas:

- Implemented the **Author** and **Reader Dashboards** for managing projects and feedback.
- Developed the **Feedback Form** to enable authors to choose what type of feedback they want their beta readers to focus on.
- Contributed to the **UI design** to create an intuitive user experience for authors and beta readers.

Collaboration:

- Worked with teammates to debug issues such as authentication and the user registration process.
- Participated in 3-5 planning sessions and standups a week.

WHAT I LEARNED

Django:

- Mastered the fundamentals of Django, including using its built-in features such as the admin panel, authentication, forms, users, models, etc.

PostgreSQL:

- Learned how to design and query relational databases for efficiency and scalability.

Full-Stack Integration:

- Improved my understanding of connecting Django’s backend with the JavaScript on the frontend.

Problem-Solving:

- Developed stronger debugging skills, especially for resolving API and database issues.

Collaboration:

- Learned how to divide tasks effectively and communicate clearly within a development team.

CHALLENGES

Learning Django:

- I had no prior experience with Django.
- **Solution**: Extensive research and experimentation to deliver the features we envisioned.

Security:

- Ensuring manuscripts were securely uploaded and accessible only to authorized users was a critical challenge.
- **Solution**: Used encrypted file storage and role-based access control.

Streamlining API Workflows:

- Creating a workflow where everything worked together, both on the author side and the reader side of the application
- **Solution**: Designed APIs to ensure smooth interaction and data flow between the author and reader sides of the application.

ACHIEVEMENTS

Functional Features:

- Delivered a fully operational Author Dashboard and Feedback Form, empowering users to manage projects and feedback effectively.

Secure System:

- Implemented robust authentication to protect user accounts and manuscripts.

User-Centric Design:

- Contributed to a polished and intuitive user experience.

Growth Through Learning:

- Successfully learned and applied new technologies like Django and PostgreSQL in a complex, real-world project.

IMPROVEMENTS - AS A DEVELOPER

Text

FUTURE DEVELOPMENT

James Sanders

THE FUTURE OF GRNLITE

- User Registration and Authentication:
- Users can sign up and log in to access the application.
- Secure user authentication to protect user accounts.
- Author Dashboard:
- Authors can manage their manuscripts efficiently.
- Upload, edit, and delete manuscripts.
- View and track the status of their submissions.
- Reader Dashboard:
- Readers can browse available manuscripts.
- Personalized recommendations or favorite lists might be available.
- Use of serializers and models to manage data interactions cleanly and securely.

THE FUTURE OF GRNLITE

- Interactive Sidebar for Authors:
- Quick navigation to important sections such as drafts, submitted manuscripts, and published works.
- Streamlined Navigation:
- Intuitive interfaces for both authors and readers.
- Organized menus to enhance user experience.
- Dynamic Features:
- Notifications or updates regarding manuscript status (e.g., approval, edits required).
- Interactive features that might include commenting, feedback, or collaborative editing.
- Backend Management:
- Django-powered backend ensures data integrity and efficient management of user and manuscript data.

Q&A

Thank you.

Team Members:

Eric Freeland - Frontend Developer
James Sanders - Backend Developer
Kelsea Conrad - Backend Developer