Discussions Are Now Available

Ivan Murashko

Contents

Introduction

Bourbaki Notes now has a discussion section below each article. The articles themselves remain static HTML and PDF files, but the discussion block is served by a small local service behind the same site. This keeps the main publication pipeline simple while making it possible to leave questions, corrections, and short technical notes directly where they belong.

The system is intentionally modest. It is not a social network and it does not try to optimize for engagement. The goal is narrower: make it easy for readers to add useful comments without requiring a third-party account or an email verification flow.

1 How Accounts Work

Readers create a local account with a username, a display name, and a password. The registration form asks for the password twice, so mistakes can be caught before the account is created:

- a username, used as a stable public identifier;

- a display name, shown next to comments;
- a password, stored only as an Argon2id hash.

No email address is collected in this first version. That is a deliberate choice. Email verification links are convenient when mail delivery is reliable, but a small VPS can have poor outgoing-mail reputation. A local account avoids that dependency and also avoids connecting the discussion system to GitHub, Google, or another external identity provider.

Logged-in readers can post top-level comments and one-level replies. The author of a comment can edit it afterwards; edited comments are marked as edited. A comment can also be reported by readers when something looks wrong.

2 Abuse Controls

The system uses deterministic checks instead of routine manual moderation. In particular, registration and early comment posting require a small browser proof-of-work challenge. The browser solves a short hash puzzle before the server accepts the action. This is designed to slow automated spam, not to prove that the reader is human.

The service also applies ordinary abuse checks: hidden honeypot fields, rate limits, duplicate-comment detection, link-count limits, and block lists for accounts, browser keys, and IP-derived keys. These checks are intentionally simple and auditable. There is no AI moderation in this version.

3 Privacy And Security Decisions

The public page shows the display name, username, timestamp, comment text, and whether a comment has been edited. It does not expose password hashes, internal session data, proof-of-work data, browser keys, or IP-derived abuse keys.

Passwords are stored with Argon2id. Session cookies are signed, HTTP-only, and kept same-site. Admin actions require CSRF tokens. The production service is configured to load secrets from files rather than raw environment variables, and the SQLite database is expected to live in a private directory with private file permissions.

The administrator can hide, unhide, or delete comments, block abusive accounts or derived identifiers, resolve reports, inspect recent abuse events, and reset password hashes. This exists for cleanup and safety, not for approving every ordinary comment.

4 Why This Shape

The main site remains a static site because that model is reliable, portable, and easy to archive. The dynamic part is deliberately kept small: it handles only accounts, sessions, comments, reports, and admin actions. This separation also makes the failure mode clear. If the comment service is unavailable, the articles and PDFs still work.

The first version is therefore conservative: local accounts, no email, no third-party login, simple proof of work, deterministic abuse controls, and a single admin interface. If the discussion area proves useful, it can grow from there without changing the way articles are written or published.