

Vaishnav M | MAGE Platform Development

My work focused on the preset platform layer of MAGE: making saved scenes easier to store, preview, organize, and retrieve through frontend and backend support. These pieces make presets easier to store, preview, organize, and connect to the user-facing parts of the web platform.

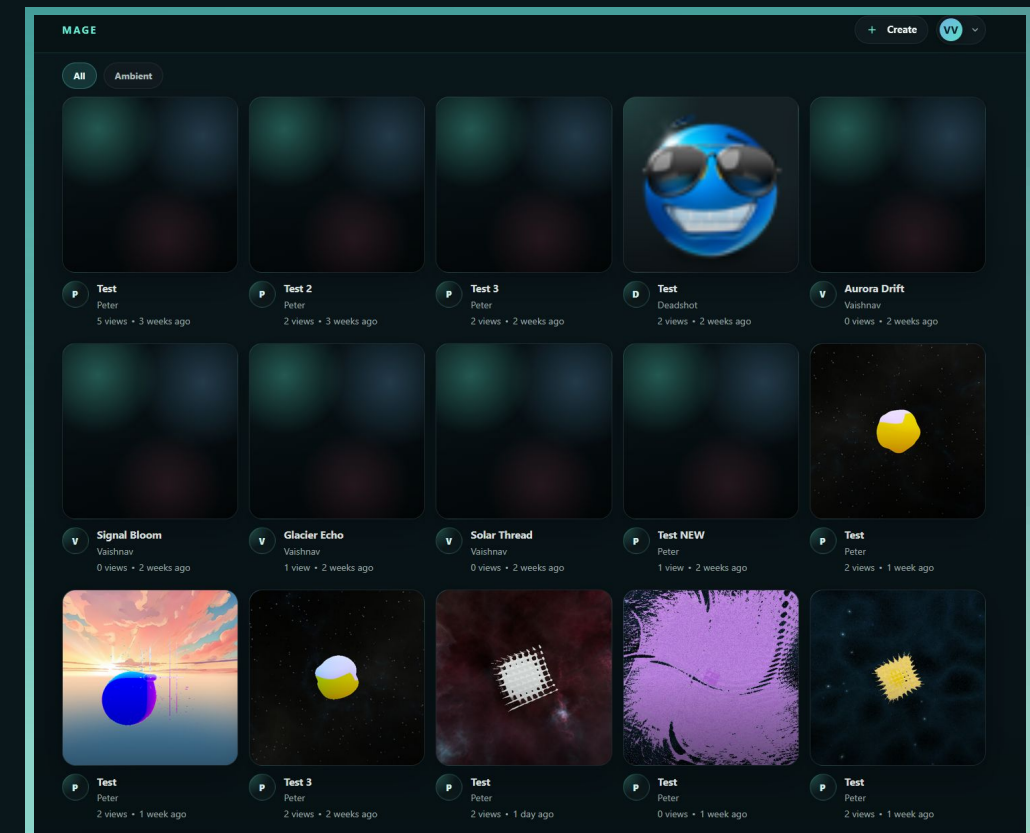
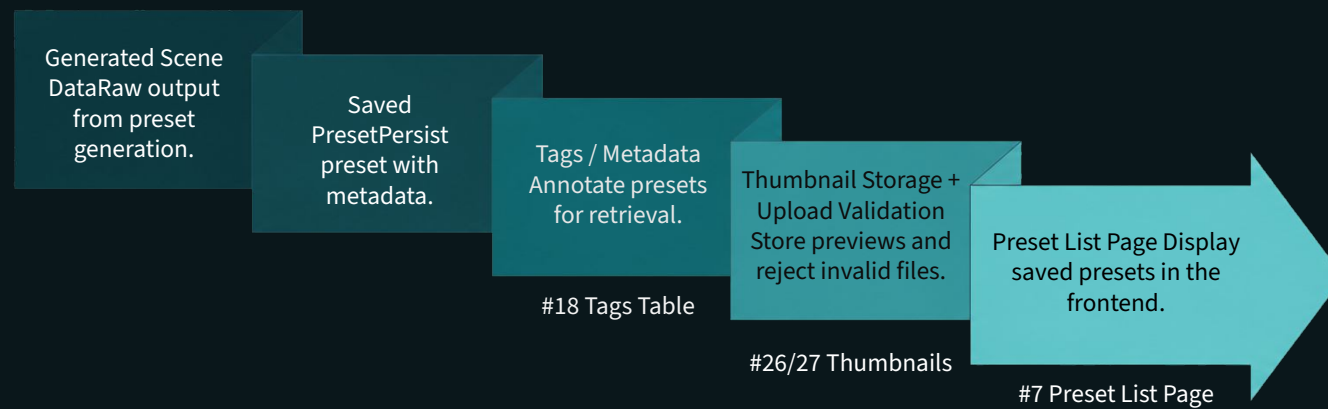
Preset list page - frontend route displaying saved presets with loading, empty, and error states

Tags database foundation - PostgreSQL schema supporting preset discovery and filtering

Local thumbnail storage - backend service saving preview images to the server's uploads directory

Upload validation - file-type and constraint checks rejecting unsafe or invalid uploads

Playlists & collections - backend domain for grouping presets into organized media libraries



My work helped move generated scene data into saved, organized, previewable, and user-facing platform features.

Resources:

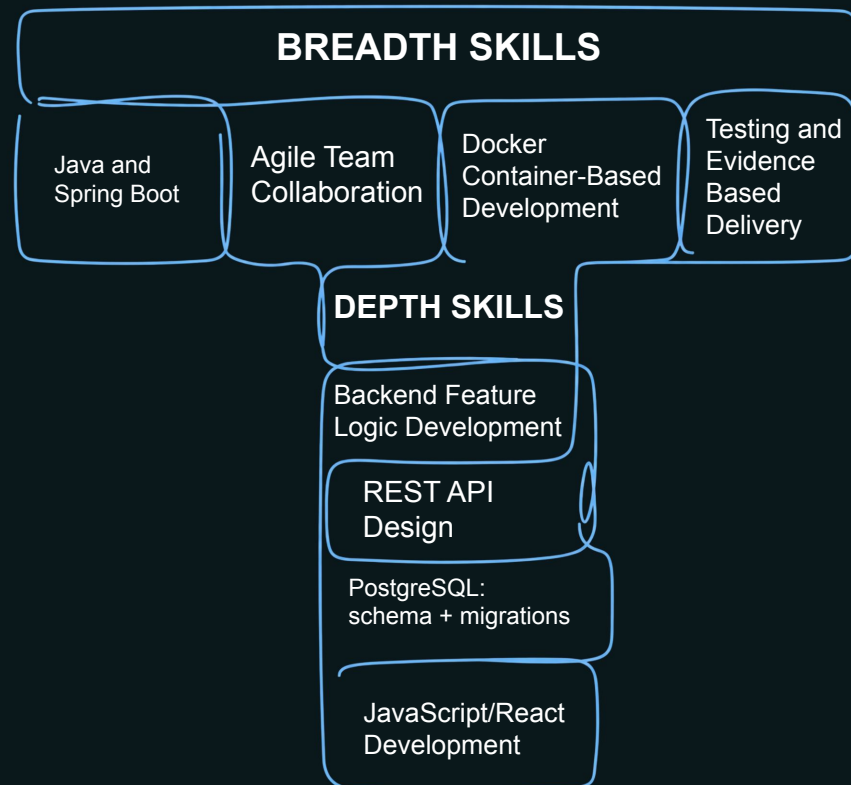
Backend repo: <https://github.com/B2MAGE/mage-backend/tree/vaishnav-backend> | **Frontend repo:** <https://github.com/B2MAGE/mage-frontend/tree/vaishnav-frontend>

Deployed site: <https://mage.peterbucci.com/> | **Frontend Contributions:** <https://github.com/B2MAGE/mage-frontend/graphs/contributors> |

Backend Contributions: <https://github.com/B2MAGE/mage-backend/graphs/contributors> | **Sprint Board:** <https://github.com/orgs/B2MAGE/projects/1>

Evidence: [#7 preset list](#) | [#18 tags](#) | [#26/#27 thumbnails](#) | [#101 playlists](#)

T-Shaped Skills: Depth & Breadth Growth



Depth

- JavaScript / React development for user-facing flows
- SQL experience with stored application data
- Backend knowledge built through prior projects

Breadth Built Through MAGE

- Java / Spring Boot service and controller patterns
- Docker-based development environment for MAGE
- Testing, PRs & Ticket Workflow

What Changed

The biggest change was learning how to transfer the skills I already had into a stack I was less familiar with. I came in with backend, SQL, JavaScript, React, and project experience, but MAGE pushed me to apply that foundation through Java, Spring Boot and also familiarize with Docker and testing environments.

Additional Comments and Next Steps

At this stage, I would focus less on adding large new systems and more on tightening the user-facing details that make MAGE feel dependable during acceptance testing.

Additional Comments

- The biggest remaining gap is not adding more features, but making the existing user flows feel dependable
- These are intentionally small next steps, but they matter because they turn visible controls into real platform behavior.
- MAGE still needs final team-level acceptance testing, not just individual feature checks.

Next Steps

Preset Interactions

- Make preset actions feel more complete and trustworthy
 - Connect visible save behavior to real user state
 - Improve how saved presets are reflected back to the user
 - Make preset browsing feel more interactive and less static
-

Account Settings

- Make profile editing persist beyond the current screen
- Connect settings changes more clearly to account identity
- Reduce local-only behavior in the settings flow
- Improve user trust and readiness for acceptance testing