



MyS EMGrowth

Prototyping

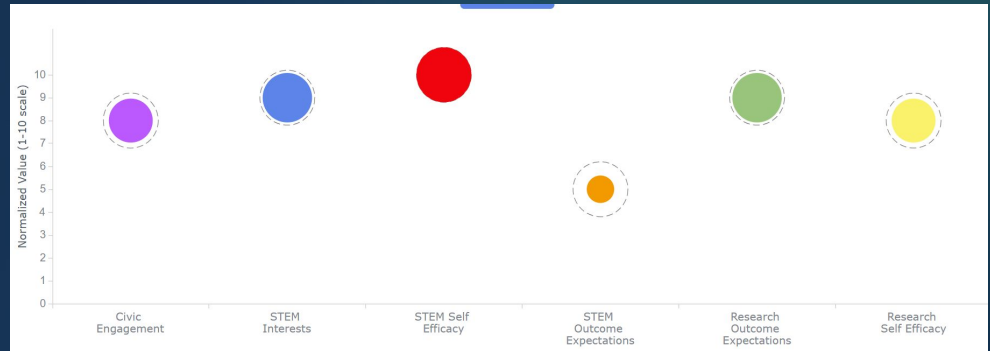
sdmay26-27

Nina Gadelha, Caleb Hemmestad, Ethan Buenting, Ethan Van Caster, Sam Craft, Ryan Mamrot



What is MySTEMGrowth?

- Web app survey measuring growth from undergraduate research
- Students complete it before and after their undergraduate research term
- Compare results to track improvement in:
 - Civic Engagement
 - STEM Interests
 - STEM Self-Efficacy
 - STEM Outcome Expectations
 - Research Outcome Expectation
 - Research Self Efficacy



Prototyping Strategy Adopted

Isolating Subsystems

- What? → Created working versions of system(s)
 - Prototype of specific features
 - User Diagrams
- Why? → Focus on individual components
 - Inherited a 'functional' tool
 - Allows us to improve on certain components (based on client proposals)
- How? → Create a user interface
 - Demo UI

Prototyping Examples: Frontend Demo UI

Enables clients to view and test different roles in an interactive environment before implementing final decisions in the live app

The screenshot shows a dashboard for 'STEM Scholars 2024'. At the top, there's a header with 'MyS EMGrowth Roles' and a 'Sign Out' button. Below the header, the main title 'STEM Scholars 2024' is followed by a sub-header: 'Coordinator view for the current STEM Scholars cohort. Use this page to keep the roster aligned, mentors supported, and resources fresh.' Below this, there's a row of five summary cards: '128 Students enrolled', '24 Mentors engaged', '43 Resources published', '92% Session attendance', and '4 Milestones this month'. A 'Switch Program' button is on the right. The main content area is divided into 'Key Workflows' and 'Today'. 'Key Workflows' includes sections for 'Student Roster', 'Mentor Support', 'Resource Library', and 'Communications'. 'Today' lists tasks like 'Confirm transportation requests' and 'Reply to 3 mentor onboarding questions'. Below this is 'Open daily checklist Helpful Links' and 'Contacts'. At the bottom, there's a 'Current Students' section with a table of student names and their last activity.

Streamlines client feedback by allowing early validation of UI layouts and feature flows in a lightweight prototype environment

Manage Programs

STEM Scholars 2024

Students
128 enrolled • 92% attendance

Mentors
24 active mentors

Next milestone
Apr 21 — Mentor training session

[View program](#) [View as student](#) [Add resource from library](#)

[Delete program](#)

STEM Scholars 2023

Students
114 alumni • archived resources

Mentors
19 mentors, 6 still engaged

Next milestone
Apr 30 — Publish year-end report

[View program](#) [View as student](#) [Add resource from library](#)

[Delete program](#)

Pathways Pilot

Students
42 participating • 3 waitlisted

Mentors
12 mentors, 4 onboarding

Next milestone
May 5 — Family info night

[View program](#) [View as student](#) [Add resource from library](#)

[Delete program](#)

Create a New Program

Start a fresh cohort with built-in templates for resources and mentor onboarding. You can tweak details before inviting students.

[Add Program](#)

Manage Resources

Create or Link a Resource

Resources live in a shared library so you can reuse them across programs. Add one here, then attach it from each cohort's Resource Library.

[Create Resource](#) [Manage library](#)

Prototyping Examples: Small Screen View

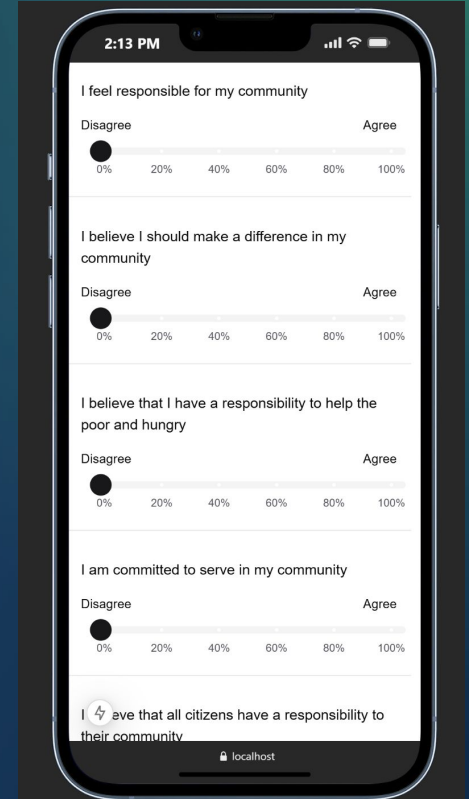
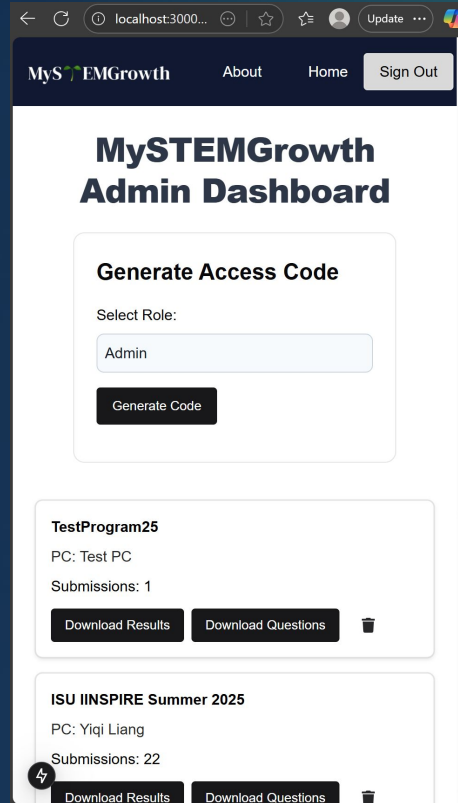
Next.js will create a development server which communicates with the backend to show changes in the front end locally without affecting the published product

```
> iinspire-app@0.1.0 dev
> next dev
```

▲ Next.js 15.1.7

```
- Local:      http://localhost:3000
- Network:    http://192.168.20.91:3000
- Environments: .env.local
```

```
✓ Starting...
✓ Ready in 3.4s
○ Compiling /admin-home ...
✓ Compiled /admin-home in 6.2s (2038 modules)
GET /admin-home 200 in 7724ms
```



Prototyping Examples: Security Improvement

- bcrypt - tool that allows for secure password storing/comparing
 - plaintext password + salt → hash
 - Never stores or compares plaintext passwords
- Prototype - demonstrates register/login flow and how passwords will be hashed and stored
 - User (plaintext) → Frontend → Backend (bcrypt: salt + hash) → Database (hashed password stored)
 - Smooth transition from prototype to production with user data security as a priority

bcrypt hashing

\$2b\$10\$ws8D6CNTKLuKrGZZpWDo/OmHkc3oC5pl.xU9XN1UkjRfepPYg1nPC

Alg → \$2b\$

Cost → 10\$

Salt → ws8D6CNTKLuKrGZZpWDo/O

Hash → mHkc3oC5pl.xU9XN1UkjRfepPYg1nPCs

```
if (req.body.password) {  
  const salt = await bcrypt.genSalt(10);  
  user.password = await bcrypt.hash(req.body.password, salt);  
}
```

```
"fullname":"gadelha","password":"$2b$10$DBU3aqcSr0cwWpdcmy0sEe4Rs7uPw3Uk9sERA0Jv4Jh3kqr/m78ee"},
```

Prototyping Examples: Cloud Services

Weighted Decision Matrix

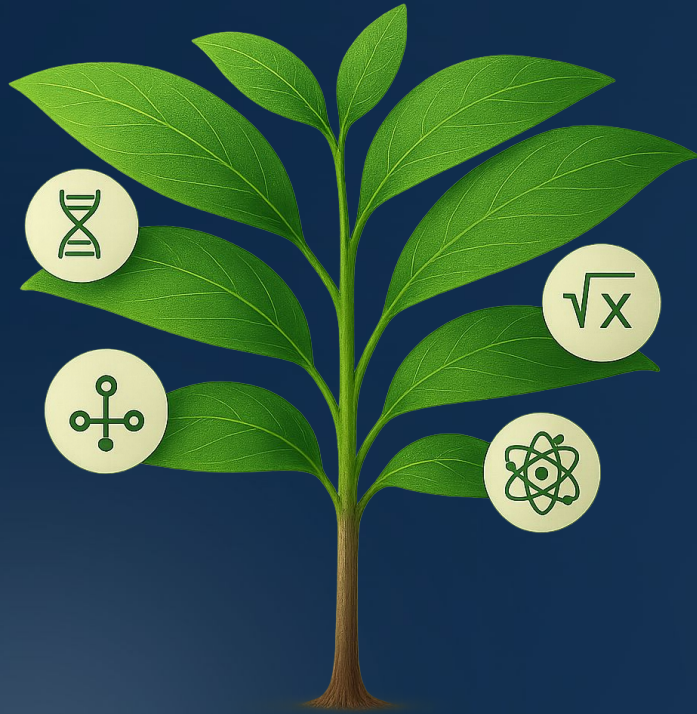
CRITERIA	WEIGHT	AWS	DIGITALOCEAN	GCP	AZURE	HEROKU/VERCEL
Cost	0.25	2	5	3	3	4
Ease of Setup	0.20	2	5	3	3	5
Integration w/ Stack	0.20	5	5	4	3	3
Security	0.20	5	4	5	5	3
Scalability	0.15	5	4	5	5	3
Weighted Score		3.7	4.7	4.2	4.0	3.8

Weights: Cost 25%, Ease 20%, Integration 20%, Security 20%, Scalability 15%. Weighted score = $\Sigma(\text{score} \times \text{weight})$.

● Outstanding (4–5) ● Competitive (3) ● Needs Work (1–2)

- Switched from AWS to DigitalOcean after evaluating performance, cost, and ease of setup
- Chose DigitalOcean for its streamlined deployment process and strong integration with our existing stack
- Now entering the beginning phases of cloud-based prototyping, testing backend workflows and scalability in the new environment

Thank you!



Questions?