SGX3CodingInstitute.html

Hackathon2024.css

Science ‘Gateways’ {

[Improving Access to HPC-ED Training Resources]

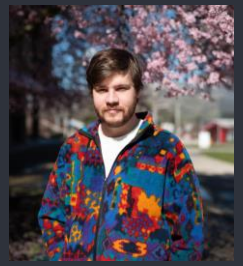
< ‘Chandler Campbell’ ‘Nole Stites’ ‘Christian Johnson’ ‘Lisha Ramon’ >

}

NLC^2.py

The ‘Faces’ Behind the Project {

< Christian Johnson
Morehouse College/SGX3
christianlj27@gmail.com >



< Chandler Campbell
Southern Oregon University/SGX3
r.chandler.campbell@gmail.com >

}

< Lisha Ramon
SUNY Oneonta/SGX3
lisharamontn@gmail.com >

< Nole Stites
Southern Oregon University/SGX3
nole.stites@gmail.com >

```
Problem Introduction {

    'Struggle to present training resources'

        <p Many institutions struggle to effectively
        categorize and present their training
        resources due to reliance on a user-unfriendly
        command-line interface (CLI). This limits
        access for users lacking CLI skills.
        Additionally, most people turn to Google for
        information, resulting in an overwhelming
        number of varied quality results, making it
        difficult to find relevant and reliable
        sources.>


        </p>
}
```

Target Science Gateway < /1 > {

< Our Targeted Science Gateway is the HPC-ED Gateway. HPC-ED (High-Performance Computing - Education) is a project to create and share metadata for HPC educational materials, making it easier to discover, access, and publish these resources through a federated catalog system. >

}

HPC ED
HPC-ED.GITHUB.IO

Goals For ‘Users’ {

User-Friendly Interface

Develop web-based platform simplifying uploading and querying data without Command Line Interface

Database Integration

Connect to a database to platform to store, retrieve, update, and delete training resources (CRUD)

Search Capabilities

Incorporate search algorithms to ensure relevant results, minimizing need to sift through irrelevant entries

User Authentication

Set up a user authentication system to ensure authorized users can add and modify resources

}

List of ‘Resource Needs’ {

- Web Development Tools — Front-end(HTML, CSS), Back-end(Python/Django)
- Authentication Providers — Globus
- Host/Deployment — Github, Docker
- User Interface — Figma, Canva

}

List of ‘Resource Needs Cont.’ {

- Team — Frontend/Backend Developers, UI/UX Designers, Instructional Designers
- Documentation Platforms — GitHub README.md, Poster/Presentation
- Collaboration Tools — Slack, Zoom

}

Hands-On ‘Resources’{

Tools

< Eureka, Django, VS Code, Figma, Linux(Ubuntu), Globus Search API >

Datasets/Testing

< HPC-ED, Dummy User Data >

}

Use Cases {

< Example scenarios >

< /1 >

* Educational Advancement:
* Universities can provide students and fauculty with easy access to HPC training materials, leading to a deeper understanding of computational techniques and fostering improved academic research.

< /2 >

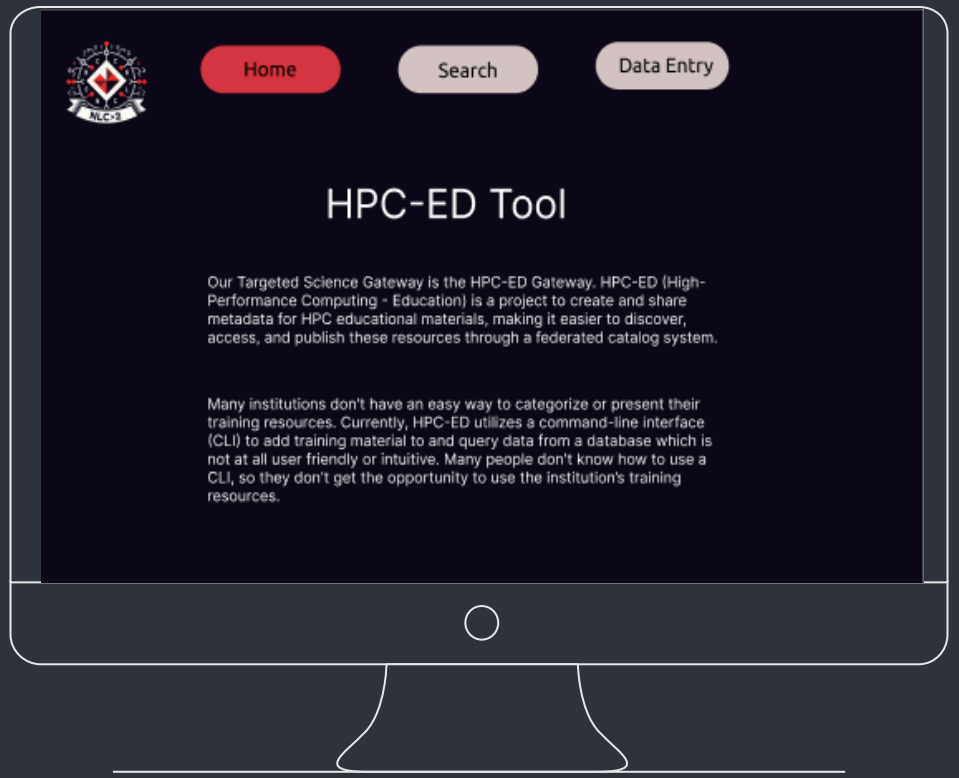
* Efficient Resource Utilization:
* Libraries and training centers can better categorize and manage their materials, ensuring users find the most relevant and high-quality resources without being overwhelmed by irrelevant information.

< /3 >

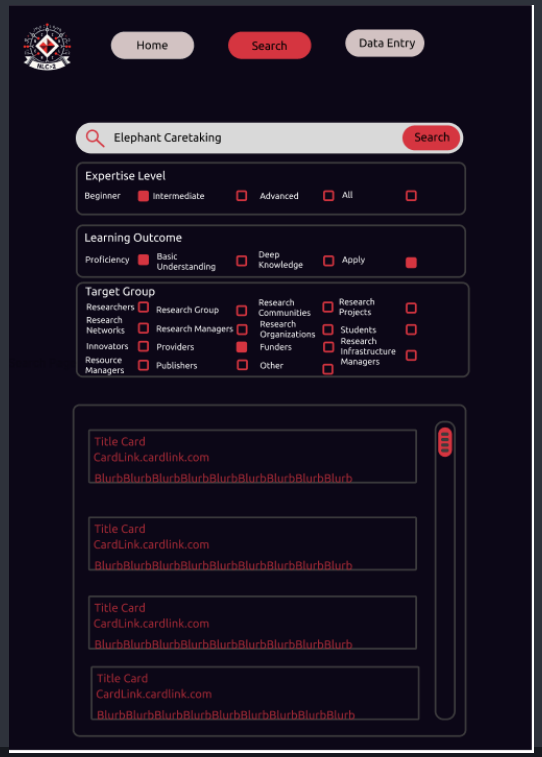
* Innovation and Research:
* Easier access to specialized training materials can spur innovation by enabling researchers to quickly learn and apply new techniques, leading to breakthroughs in various scientific fields.

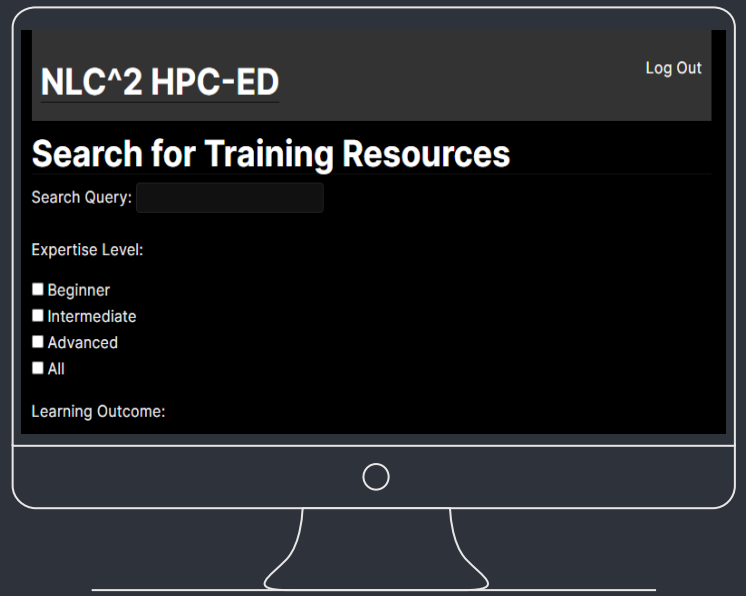
}

```
1
2
3
4  Home User Interface {
5
6
7
8
9
10
11
12
13
14  }
```



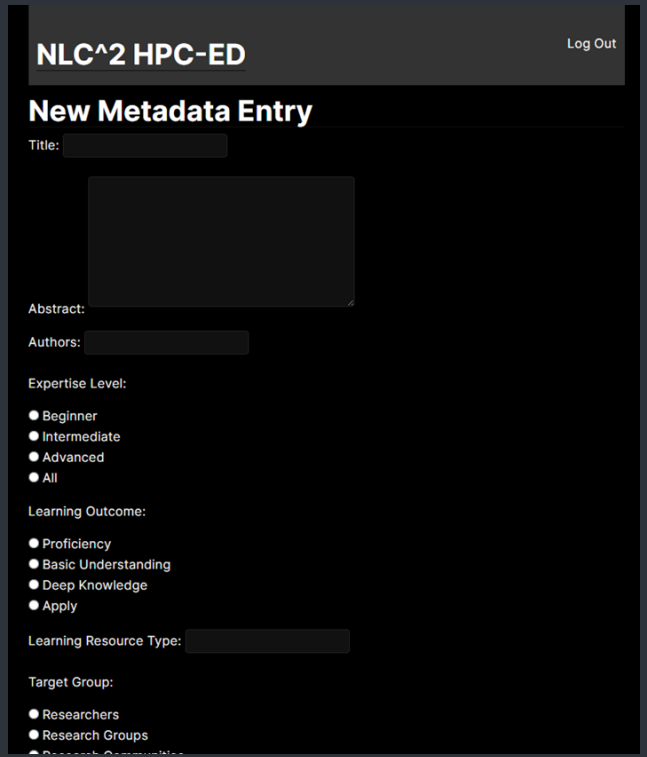
Search User Interface {

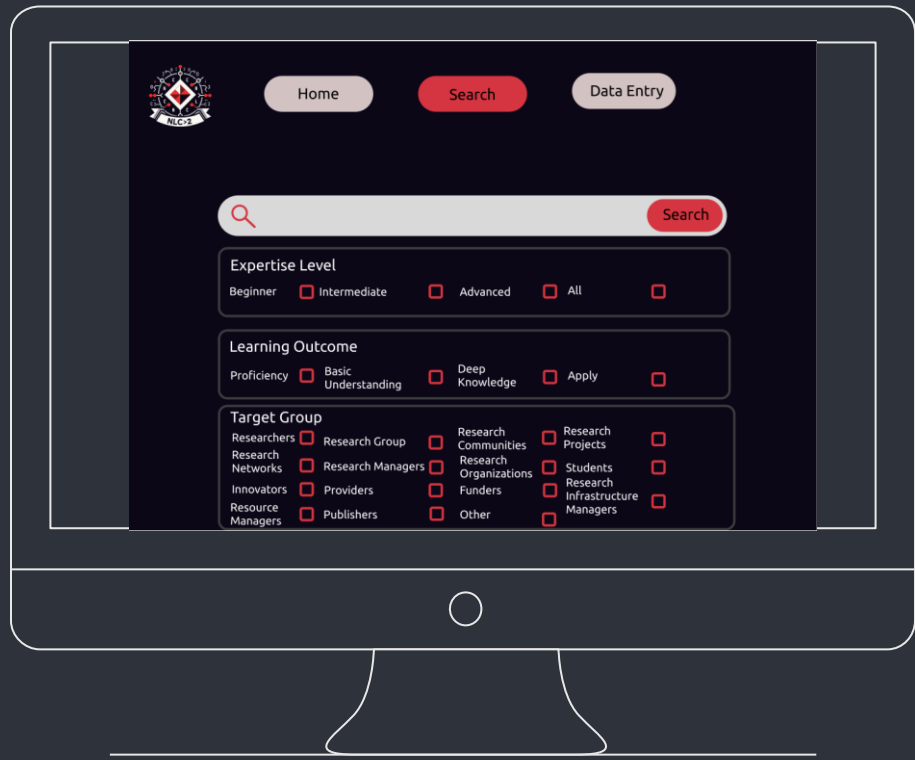




}

```
Data Entry Interface {
}
```





```
Sign In Interface
{
}
```

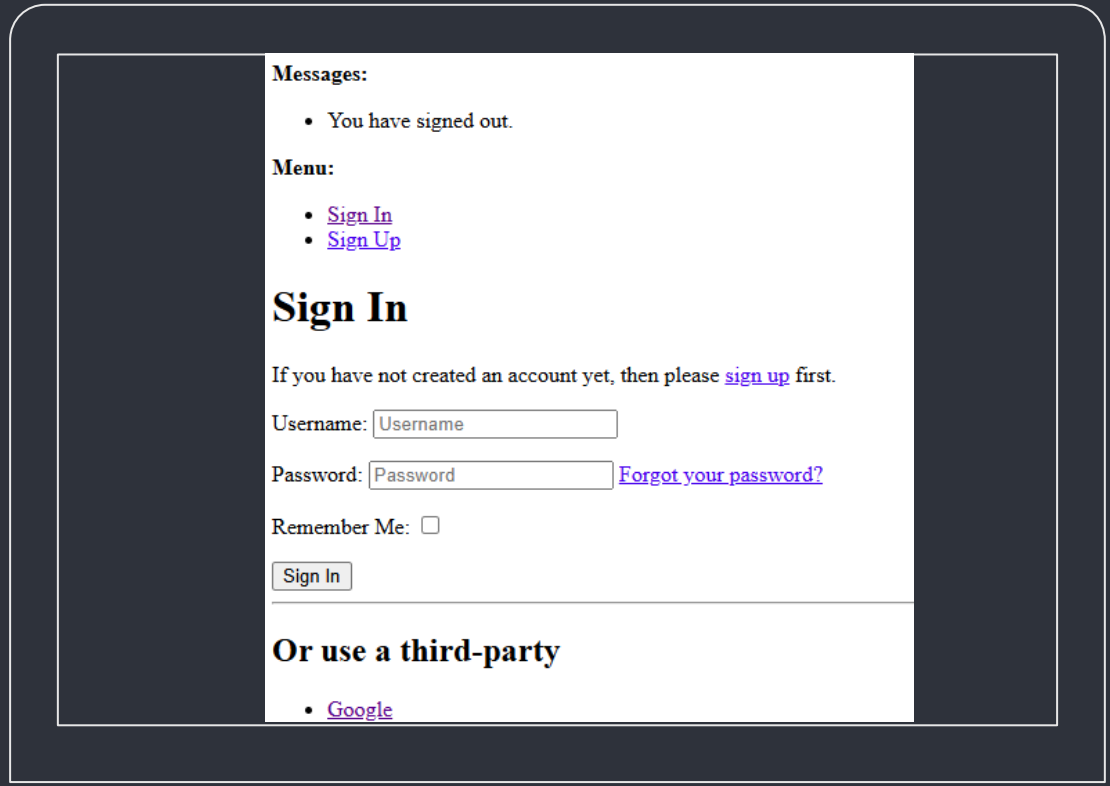


Table Of ‘Methodology’ {

01 Planning and Requirements Gathering

< All members helped in defining project goals, gather user requirements, and create a timeline to establish important milestones. >

02 Design Phase

< Design team created wireframes based on user requirements which then were developed into UI designs for user-friendly experience. >

03 Front-End Development

< Developers converted the wireframes into HTML/CSS, implementing interactive parts and ensuring navigation. Also, integrated frontend with backend services using APIs. >

}

Table Of ‘Methodology Cont.’ {

04 Back-End Development

< Team set up the server environment using Python with Django. Developed APIs for CRUD [create, read, update, delete] operations. >

05 Integration and Testing

< The front-end integrated with the back-end through APIs. Comprehensive integration testing was performed as well as usability testing to identify performance issues. >

06 Deployment

< Deployment team set up the production environment using Django on a web hosting platform. >

}

Future Enhancements; {

Improved Search Capabilities

< To generate unique list of filters based on metadata in database entries, allowing for further refined searches. >

Enhanced Authentication

< Addition of Google Authentication would provide extra security, making platform more secure. >

Responsive Design

< Ensuring the platform is fully responsive and accessible on various devices, including desktops, tablets, and smartphones. This would cater to a wider range of users. >

}

01 {

[Community Impact]

< By choosing to address this problem, we are able to enhance community access to high-quality training resources. In turn, will lead to improved education and skill development, foster collaboration and knowledge sharing, and ultimately driving innovation and progress within various fields. >

}

< “Science gateways bridge the gap between advanced computational tools and the scientists who need them, enabling groundbreaking research and innovation.” >

— Vinton Cerf, ‘co-inventor of the Internet and Chief Internet Evangelist at Google’

```
Thank You; {
    ‘Do you have any questions
    for us?’
}
```

< Please speak now or forever hold your peace >