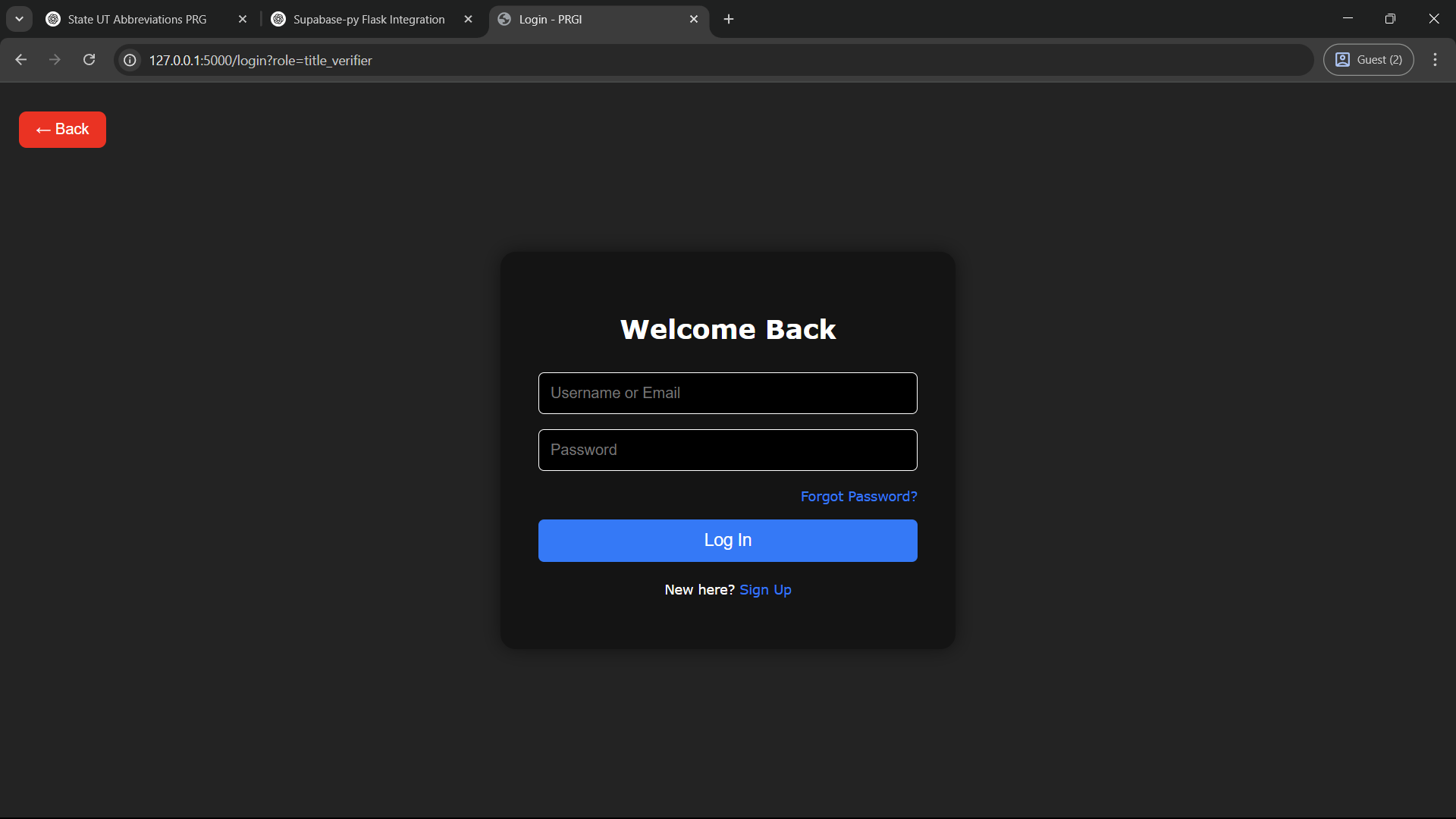
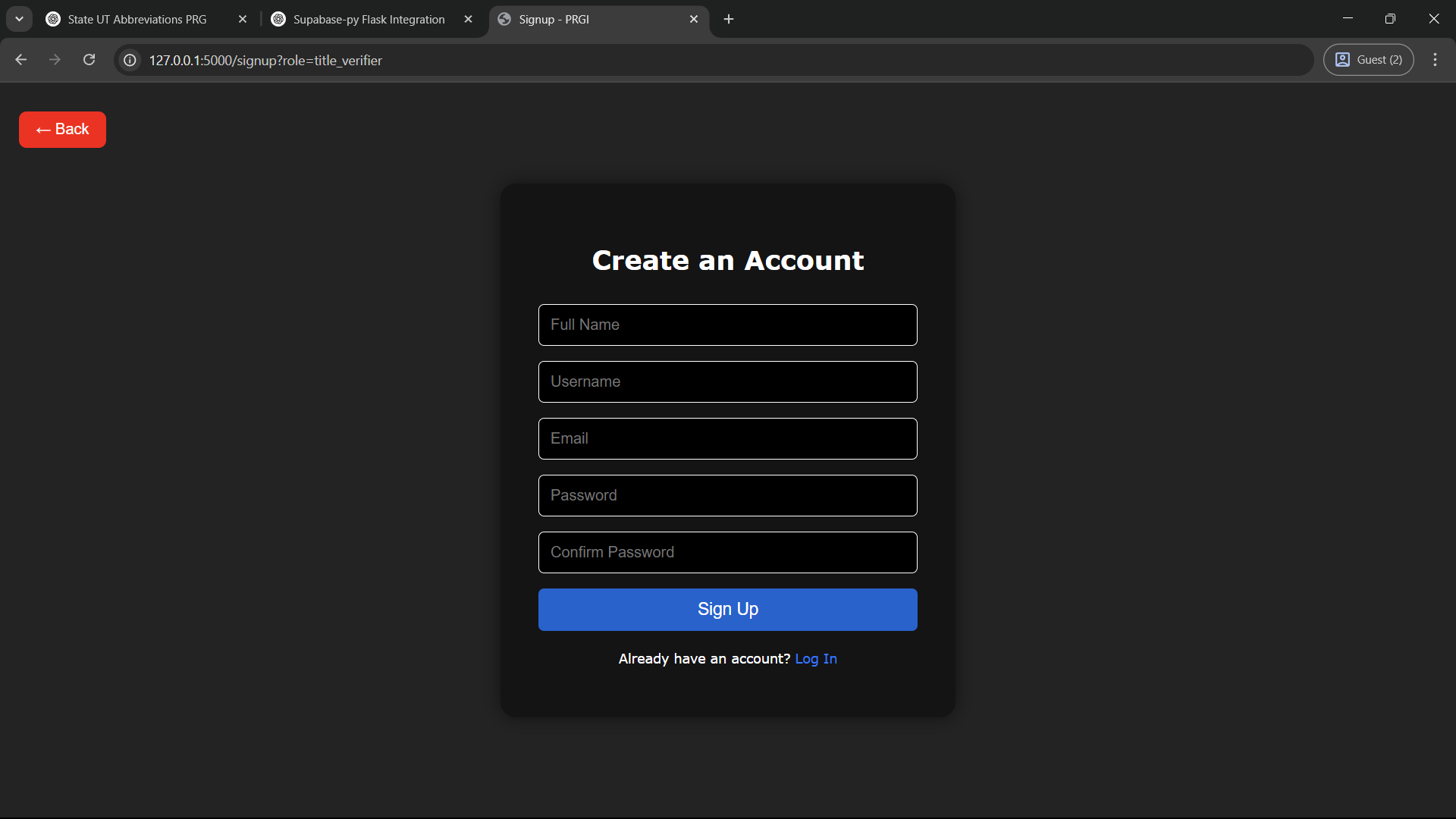
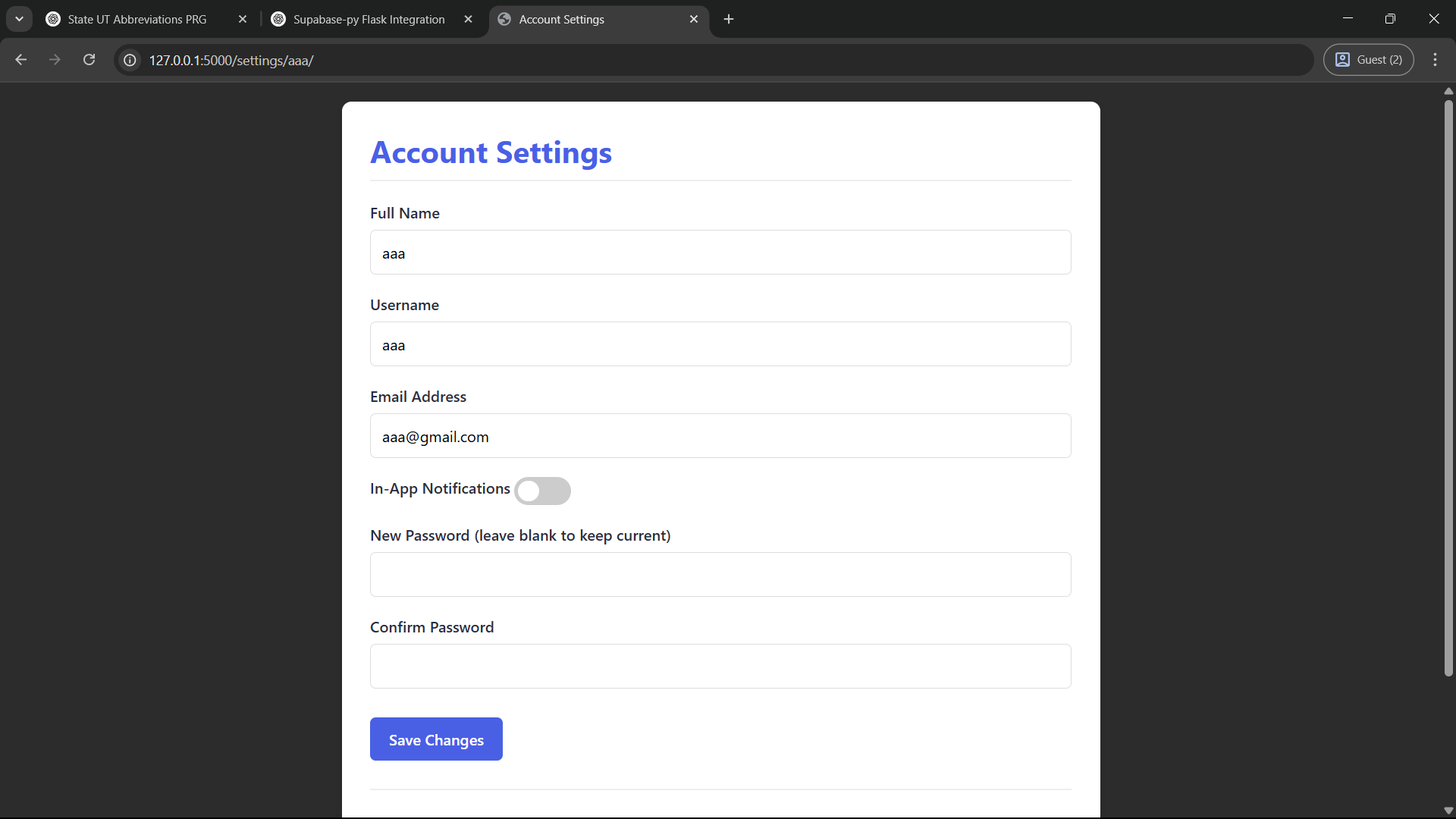
**Log in**



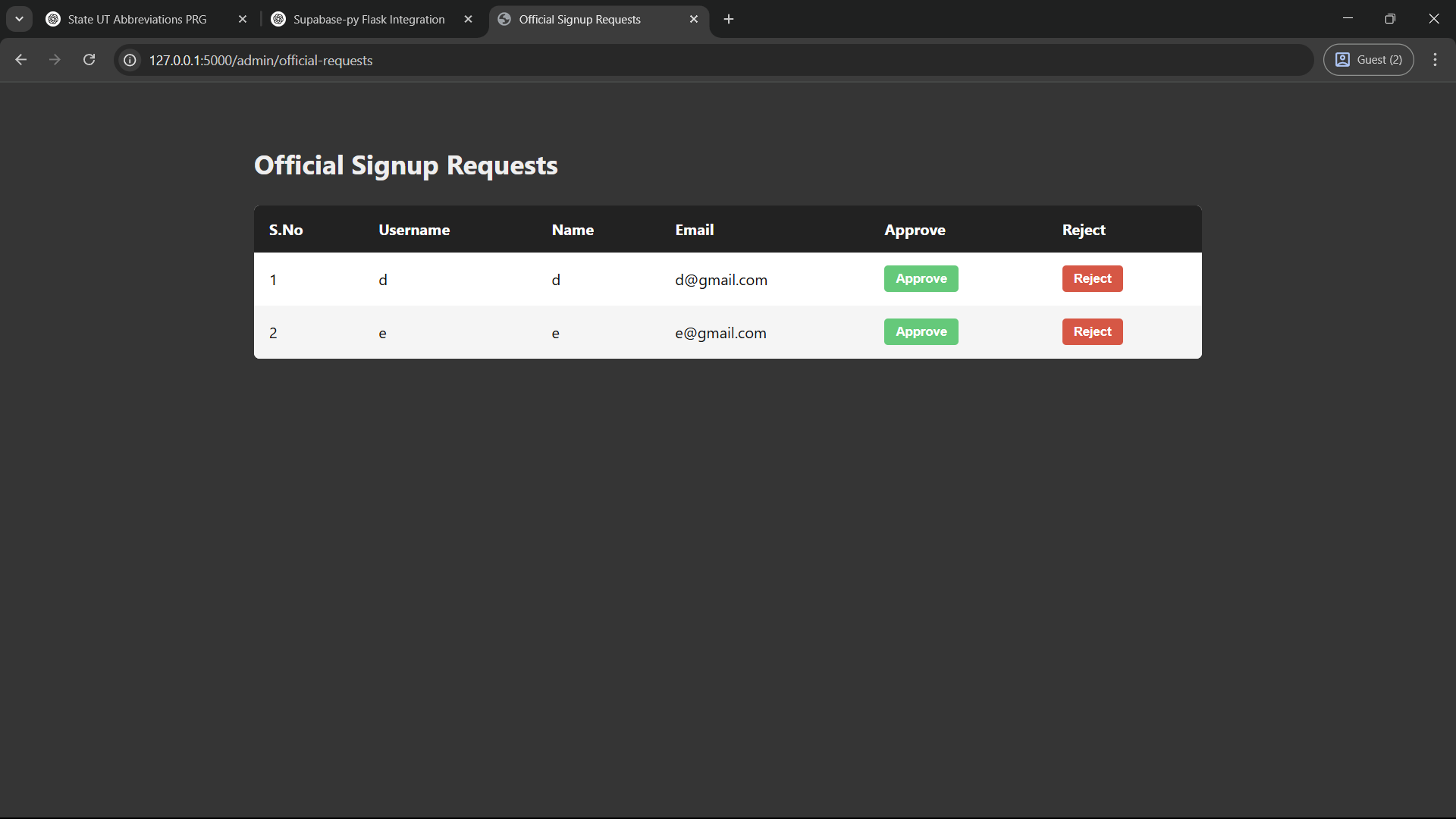
**Sign up**

****

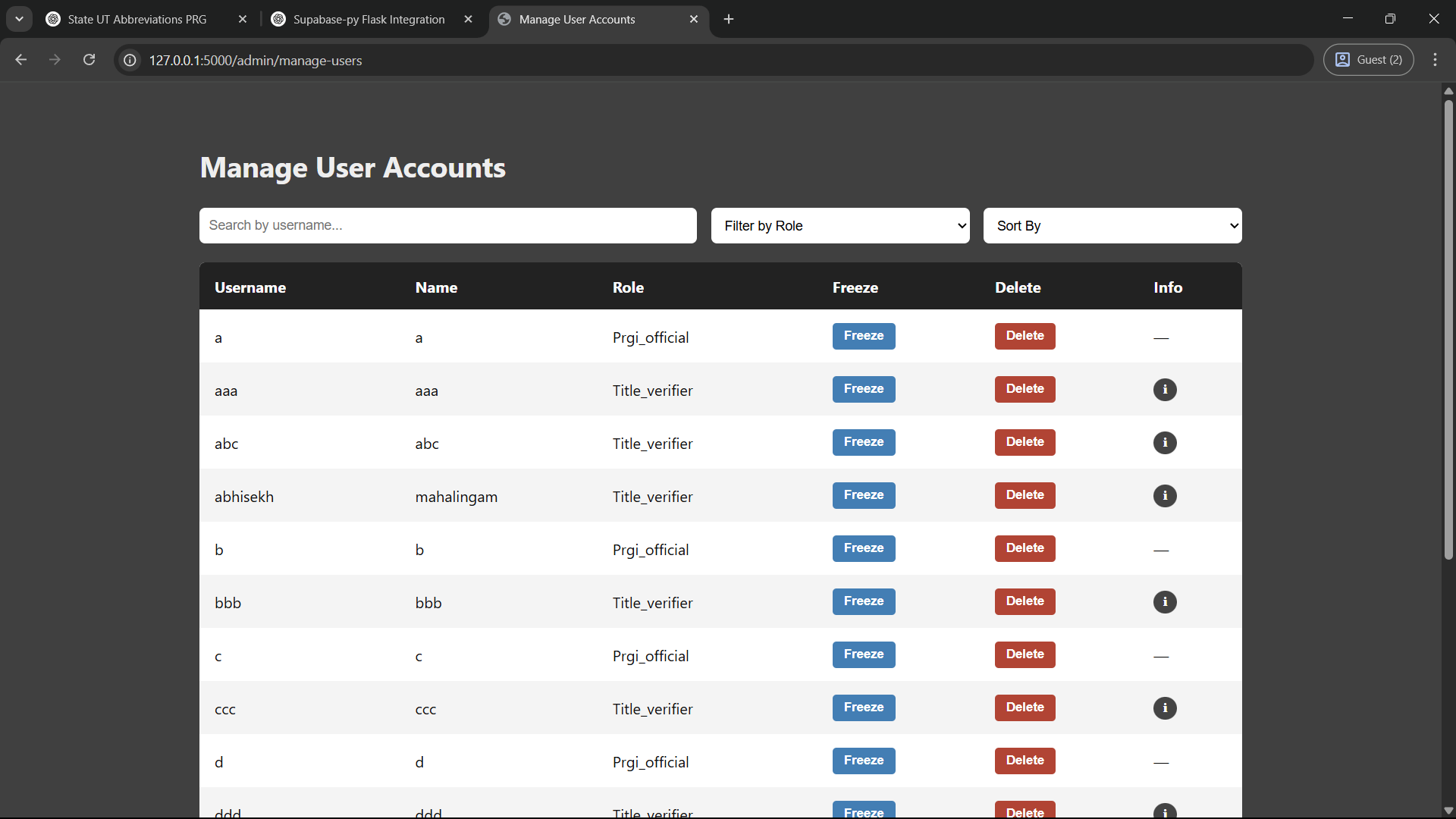
**Manage Account Settings**

****

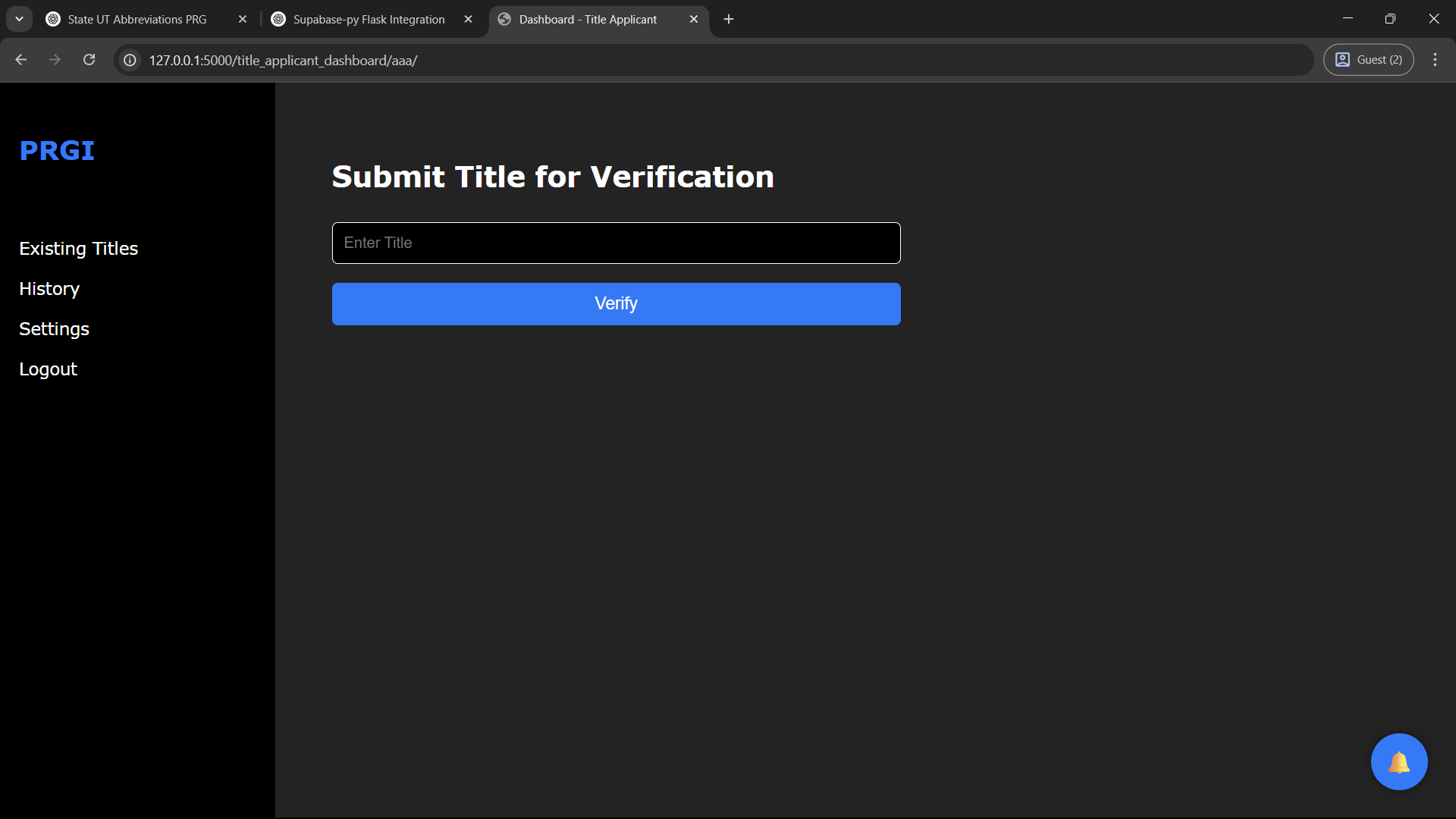
**Approve/Reject Official Accounts**

****

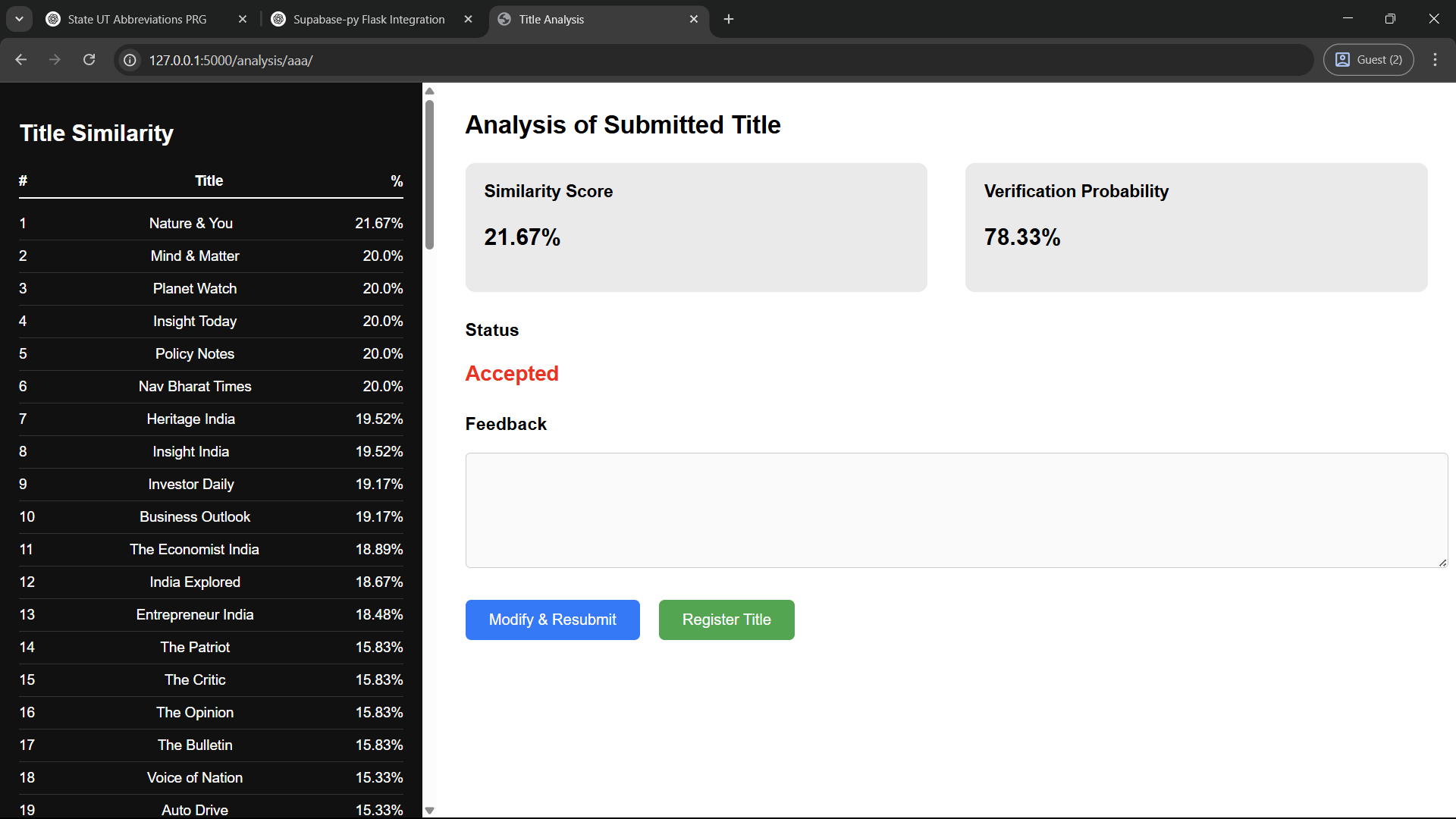
**Manage User Accounts**

****

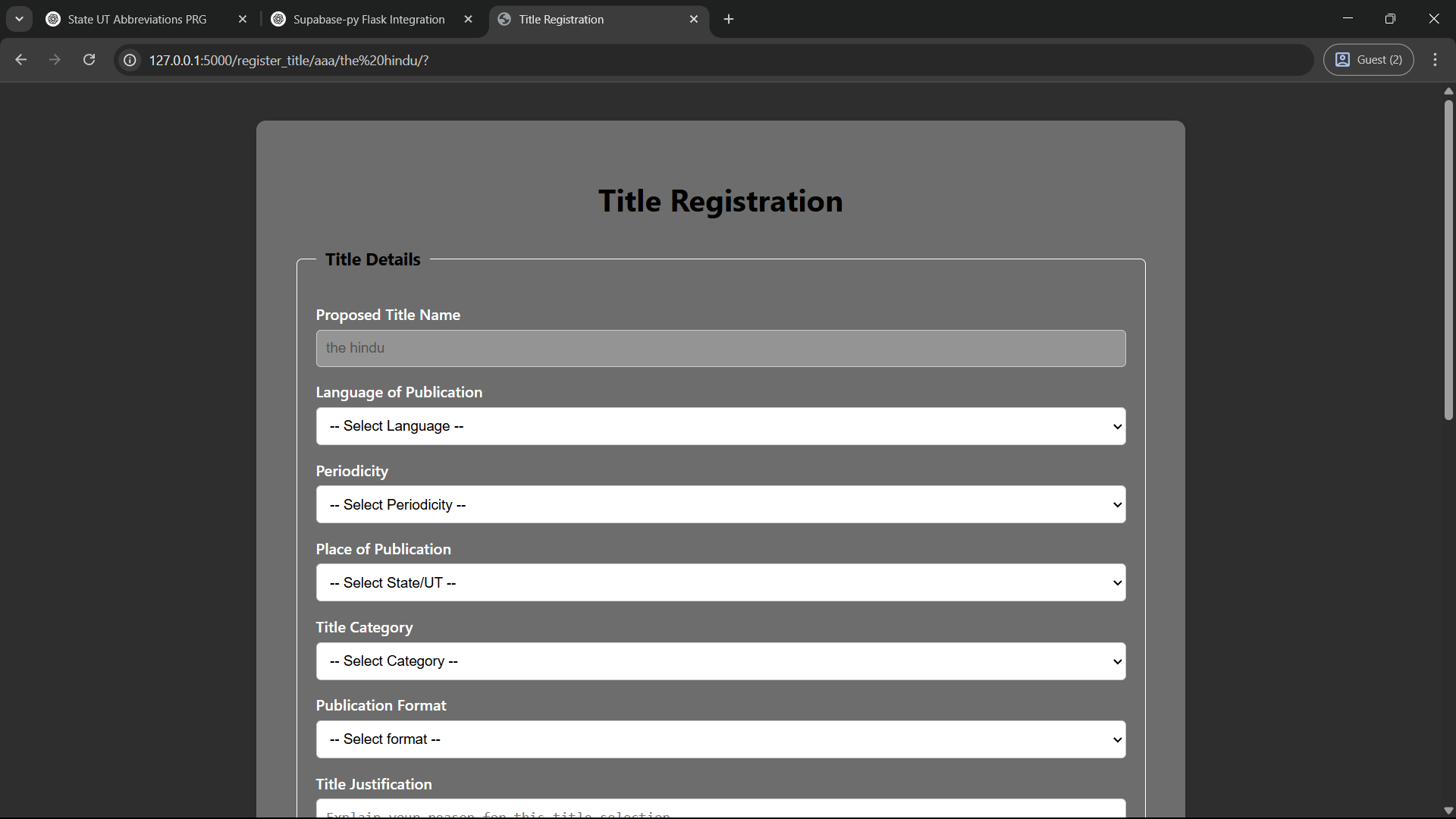
**Submit Title for Verification**

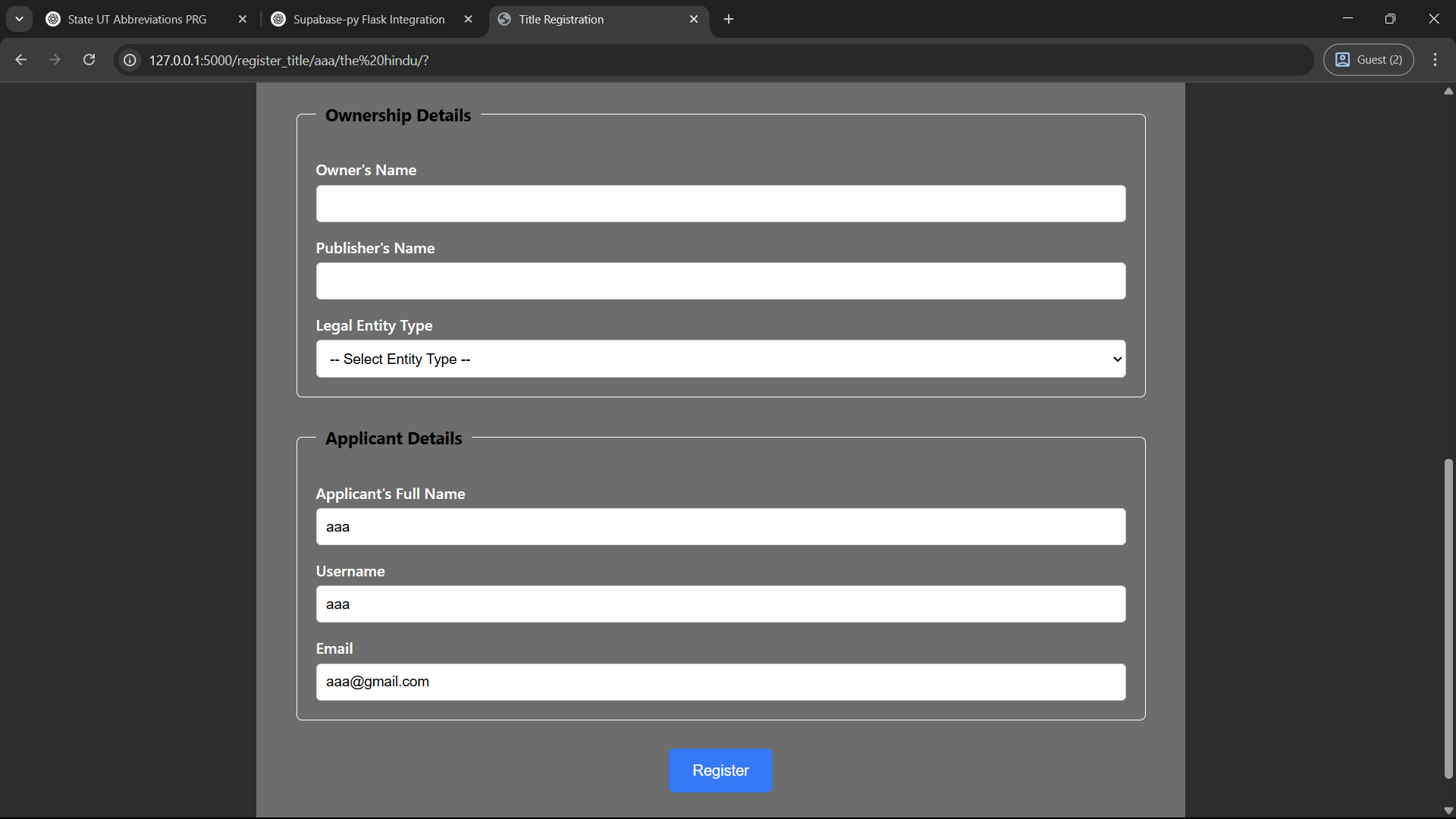
****

**View Title Analysis and Result**

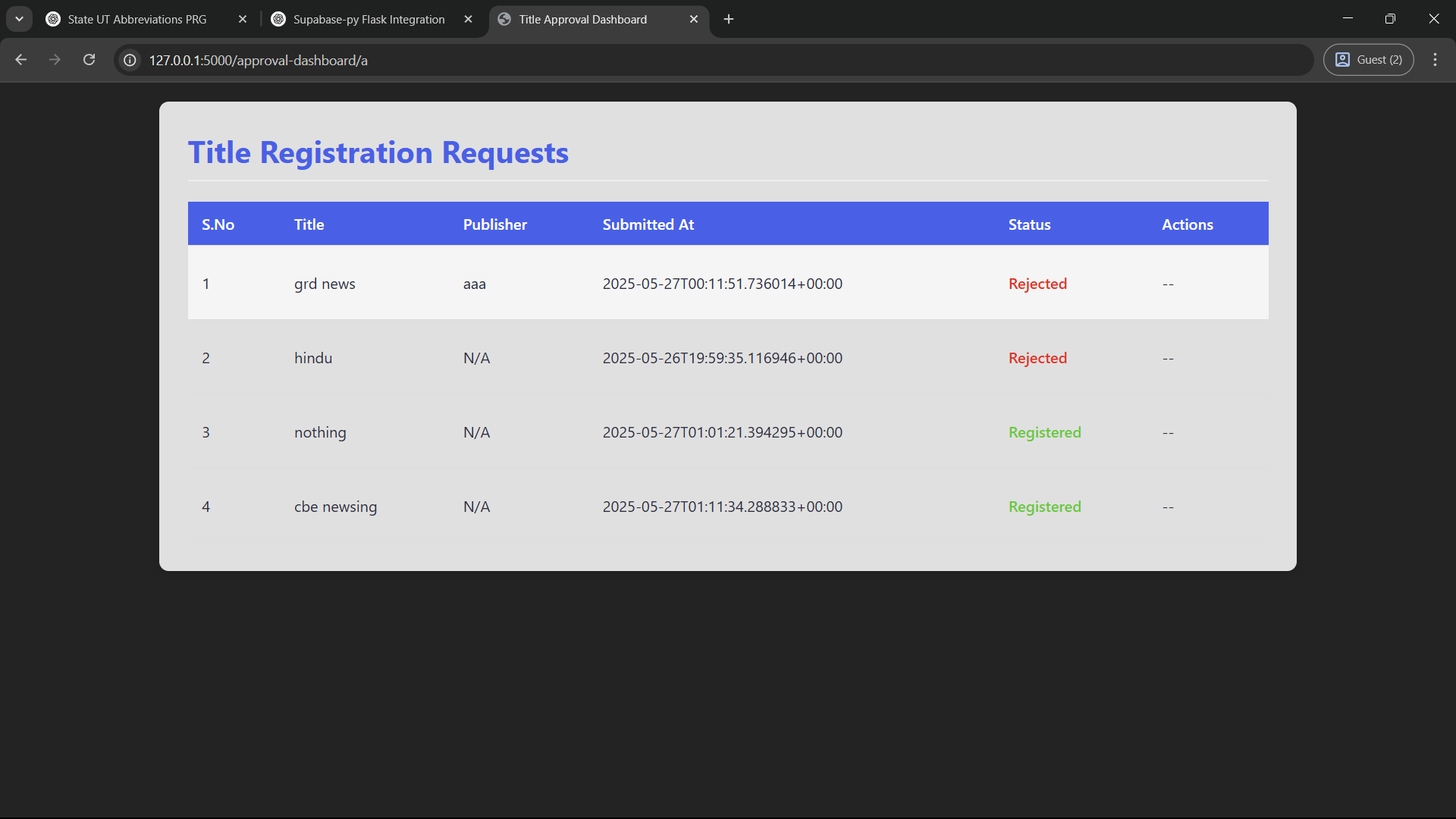
****

**Register Title**

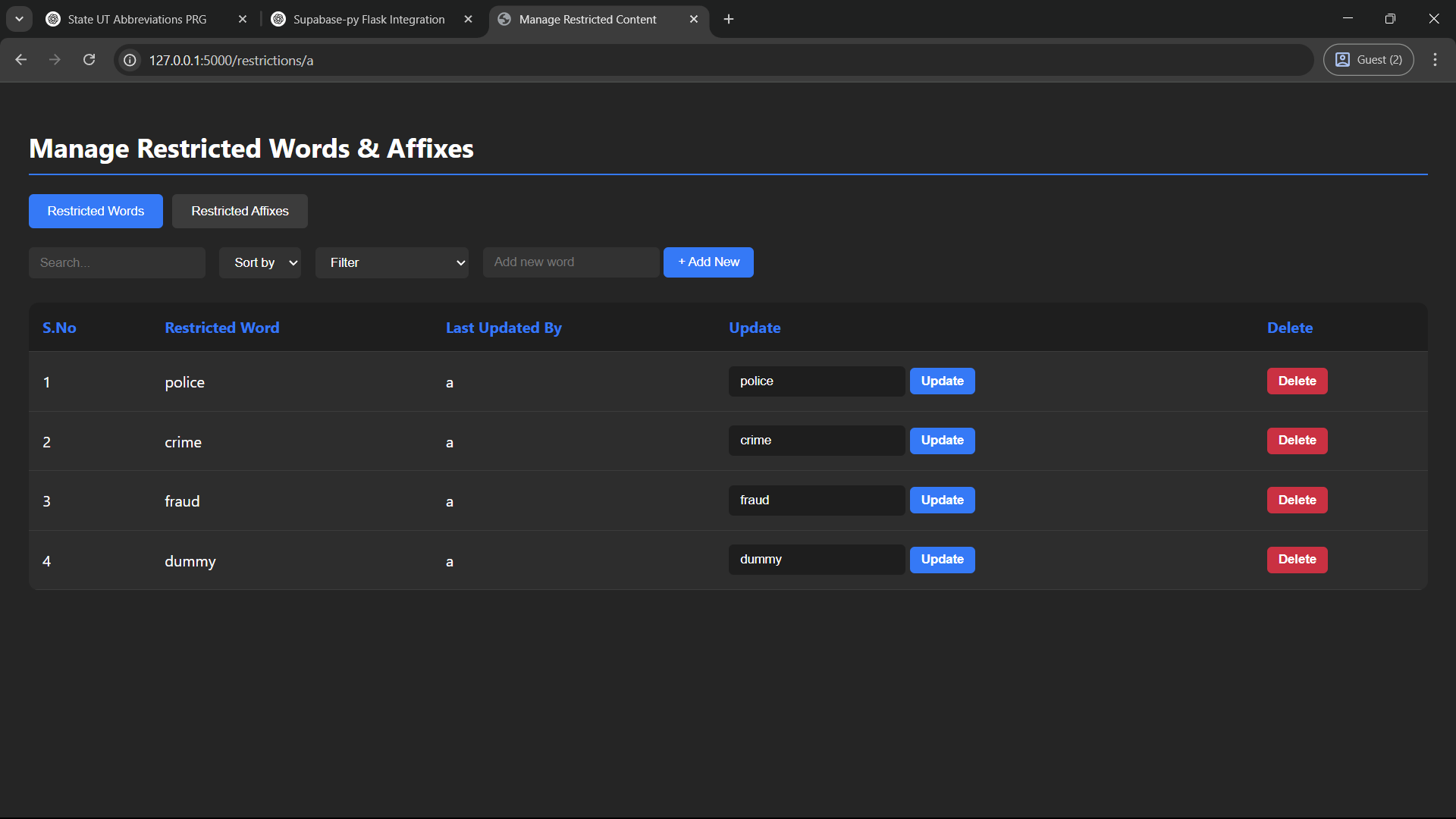
****

****

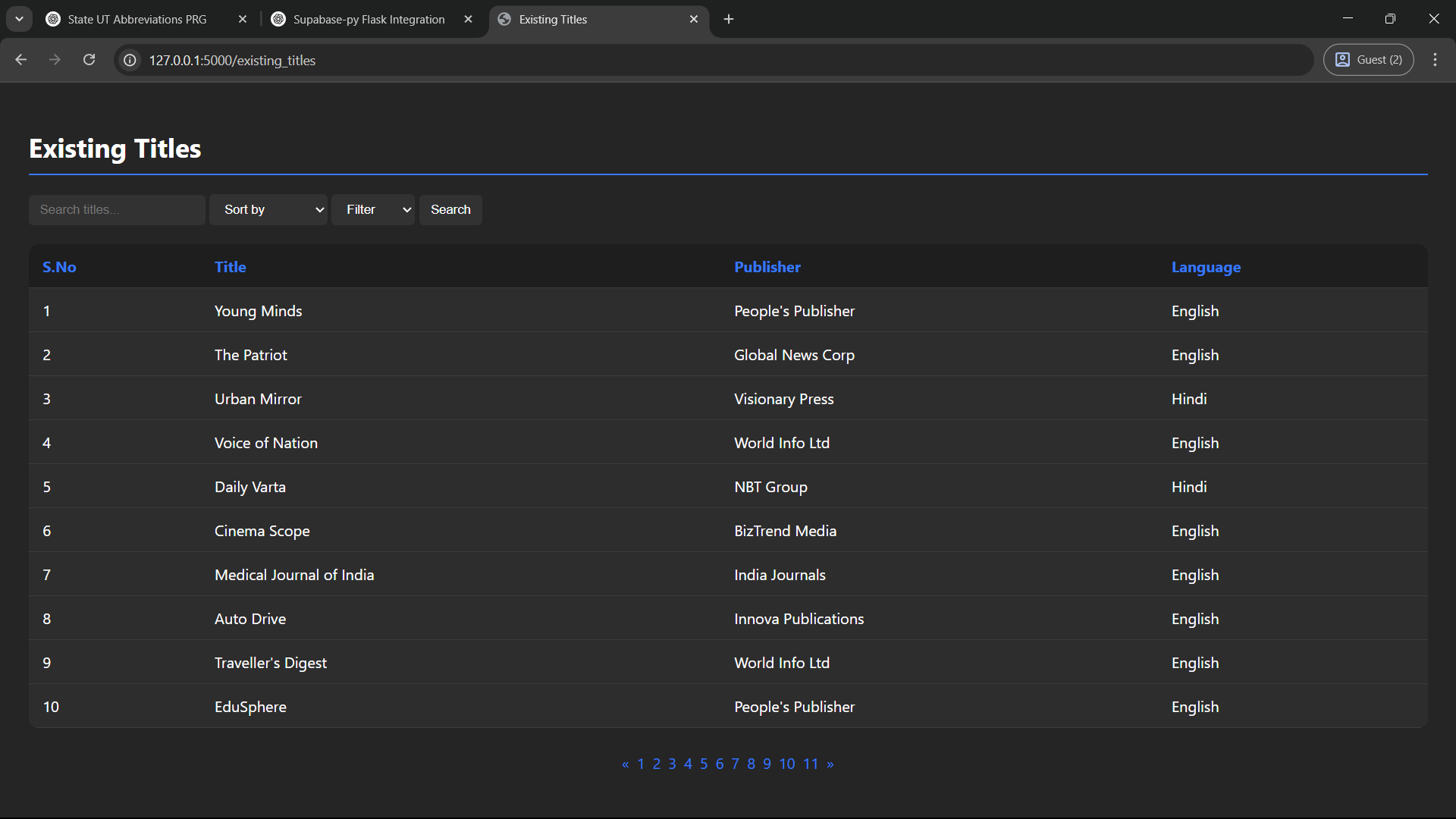
**Approve/Reject Title Registrations**

****

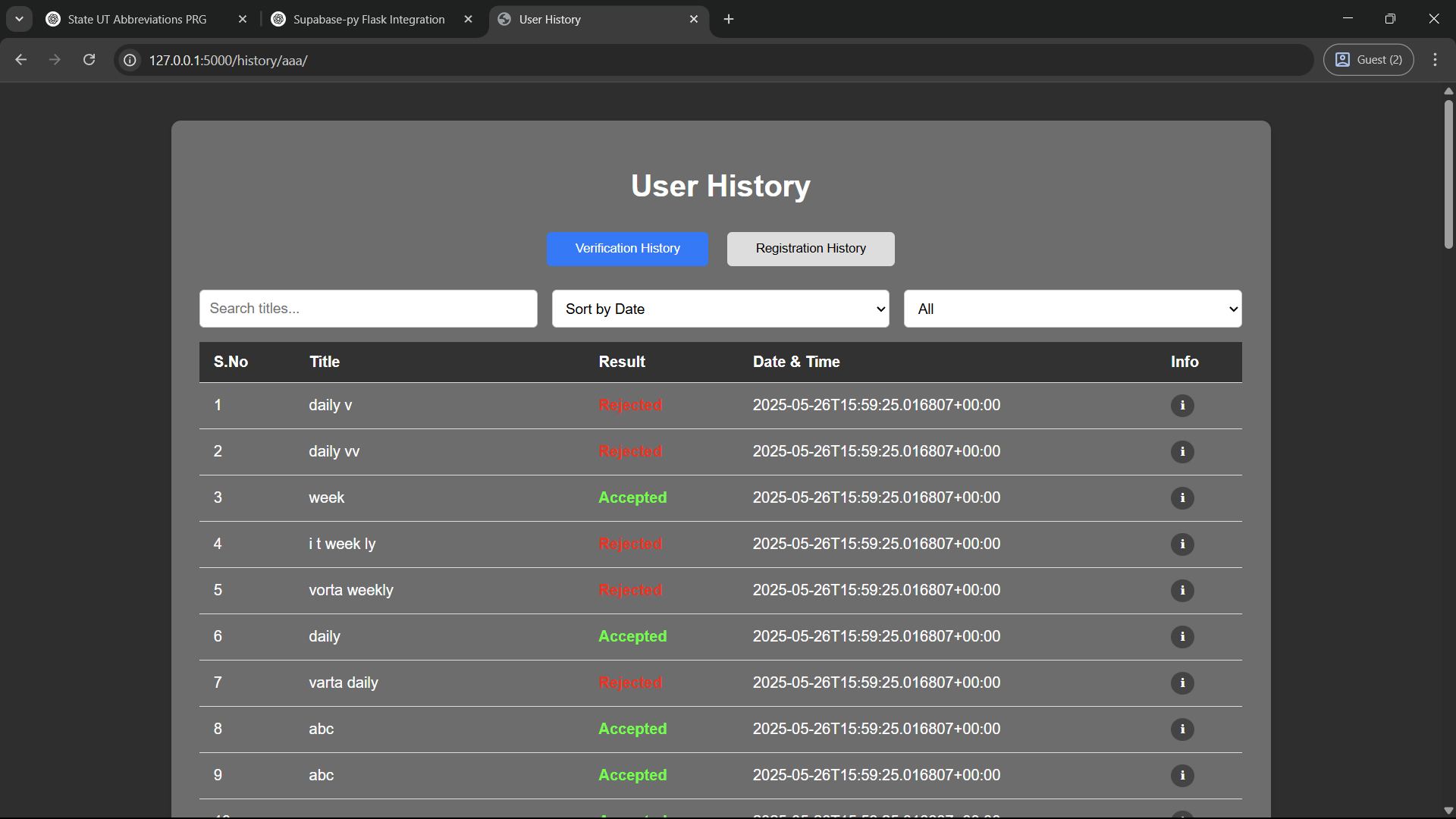
**Manage Restricted Words and Affixes**

****

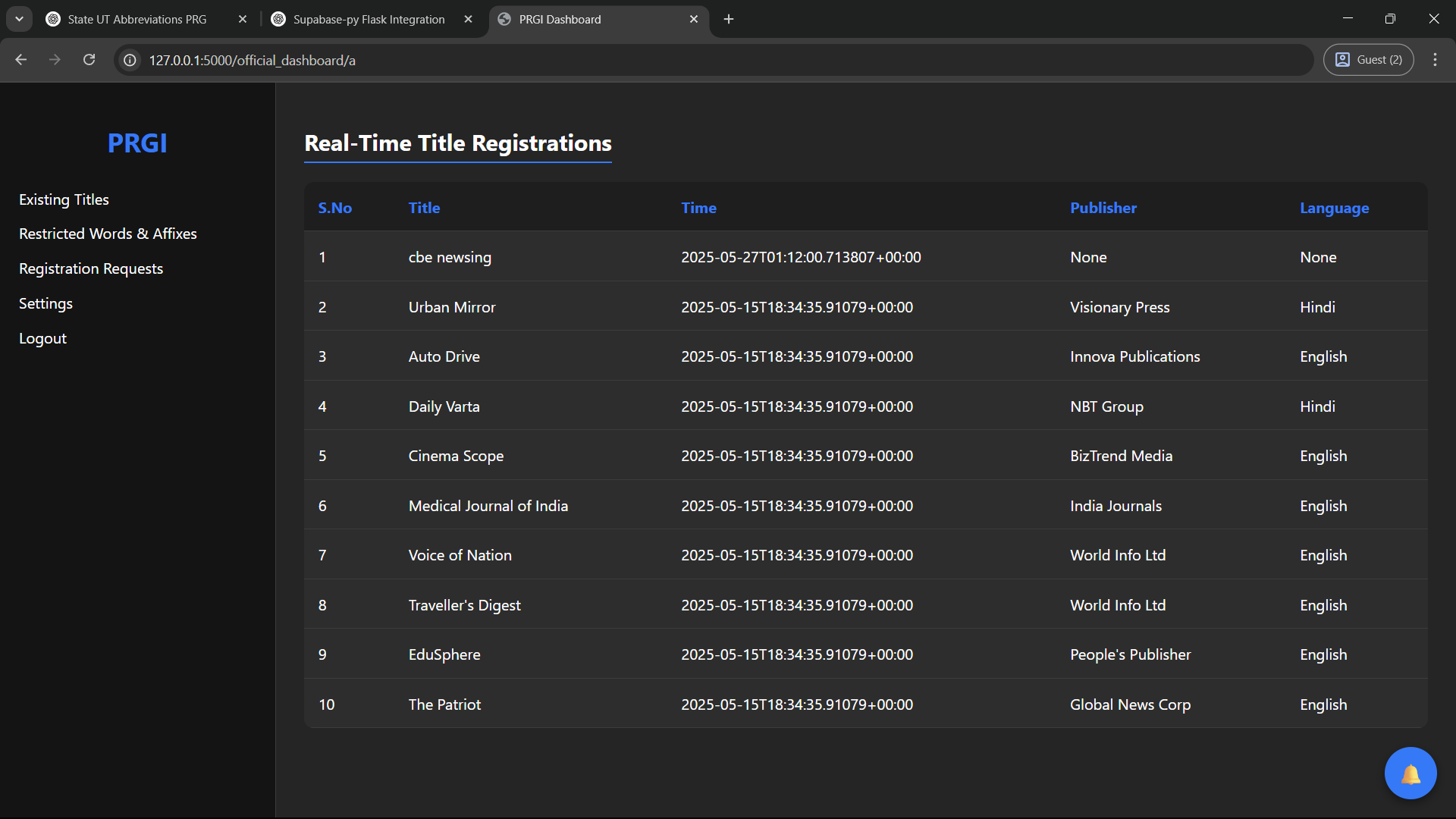
**Search and View Existing Titles**

****

**View Submission History**

****

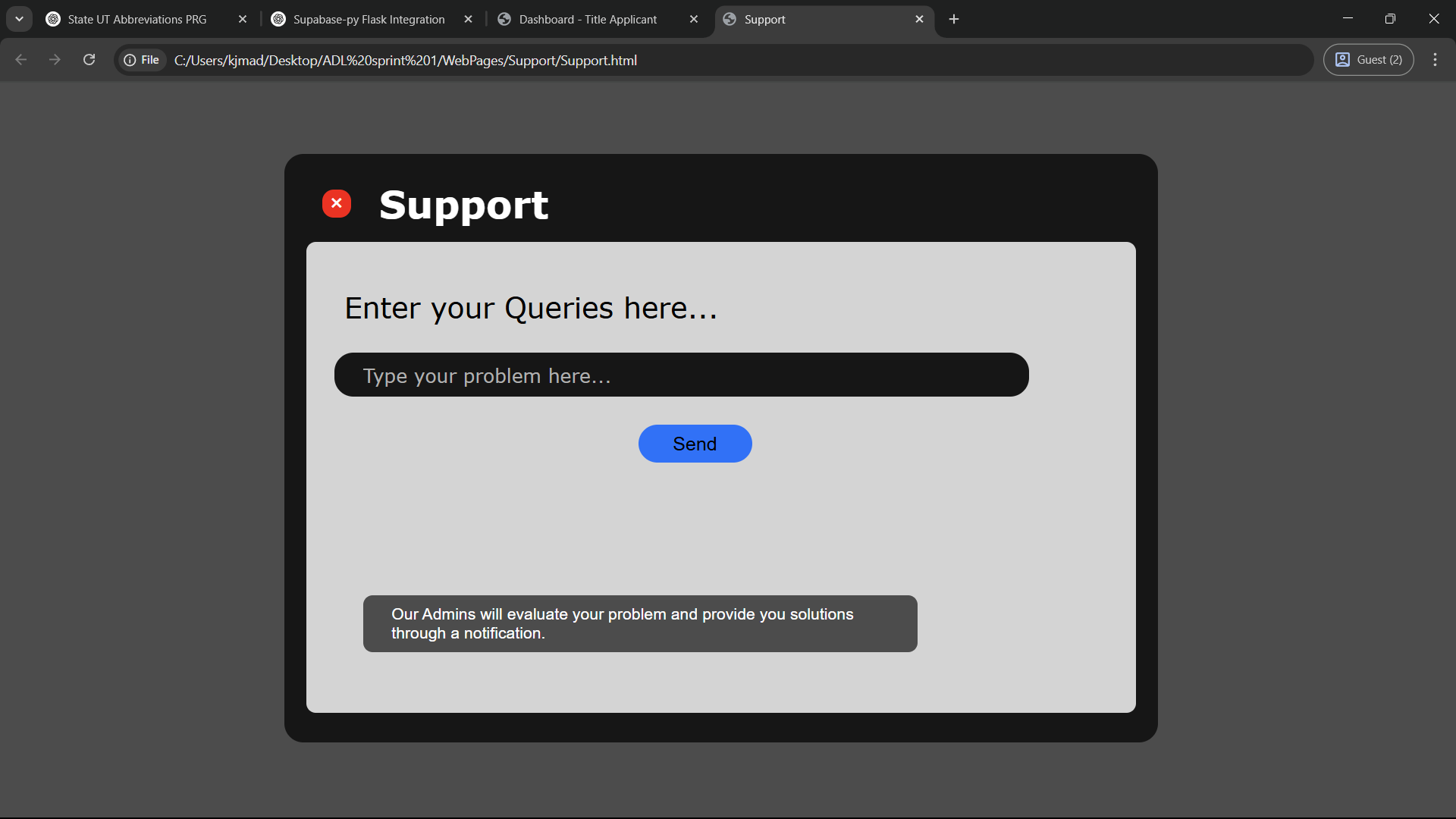
**Monitor Title Registration History**

****

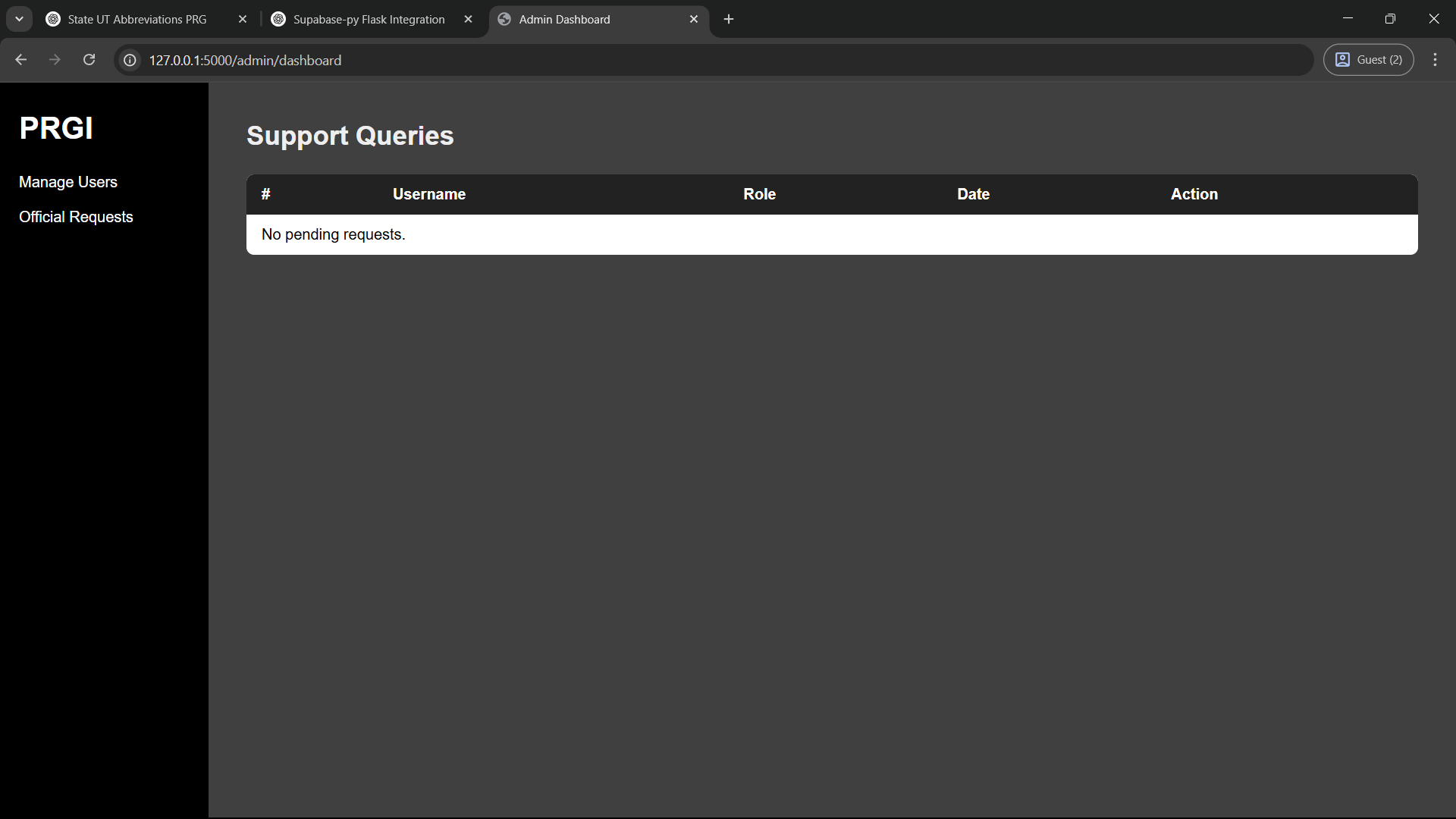
**Access Help & Guidelines**

**-**

**Contact Support**

****

**Respond to Support Queries**

****

**Software Requirements**

This project is developed using a web-based architecture with a structured and scalable technology stack. The selected tools and libraries are optimized for performance, modularity, and ease of learning.

**1. Frontend**

* **Technologies**: HTML, CSS
* **Description**: The frontend is developed using standard HTML for structure and CSS for styling. It includes user interfaces for title submission and administrative controls.

**2. Backend**

* **Technology**: Python with Flask Framework
* **Description**: Flask is used as the backend framework for handling routing, form submissions, title verification logic, and database interactions. It provides a minimal and flexible foundation suitable for building web applications and APIs.

**3. Database**

* **Technology**: PostgreSQL (Hosted on Supabase)
* **Description**: PostgreSQL is used as the primary relational database to store titles, restricted words or affixes, user records, and logs. Supabase offers a managed cloud environment for hosting the PostgreSQL instance, providing scalability, security, and administrative tools.

**4. Libraries for Similarity Checking**

* **Libraries Used**:
  + sklearn.feature\_extraction.text.TfidfVectorizer: Used for vectorizing text data and calculating semantic similarity through cosine similarity.
  + metaphone (from the fuzzy or similar phonetic library): Used to encode phonetic representations of words for comparing pronunciation similarity.
  + jaro\_winkler\_similarity (from jellyfish or equivalent): Used for computing phonetic similarity scores between encoded titles.
  + Python standard operations (set, split, string preprocessing): Used for calculating spelling similarity through Jaccard similarity and for stopword filtering.
* **Purpose**: These tools are used in combination to compare submitted titles with existing ones using a multi-dimensional similarity approach, incorporating spelling, phonetic, and semantic matching. The final similarity score determines whether a title is unique enough to be registered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case No.** | **Use Case Name** | **Test Case No.** | **Status** | **Screenshot** |
| TS1 | Sign Up | TC1 – TC14 | Pass | – |
| TS2 | Log In | TC1 – TC16 | Pass | – |
| TS3 | Manage Account Settings | TC1 – TC14 | No Run | – |
| TS4 | Approve/Reject Official Accounts | TC1 – TC11 | Pass | – |
| TS5 | Manage User Accounts | TC1 – TC14 | Pass | – |
| TS6 | Submit Title for Verification | TC1 – TC10 | Pass | – |
| TS7 | View Title Analysis and Result | TC1 – TC11 | Pass | – |
| TS8 | Register Title | TC1 – TC12 | Pass | – |
| TS9 | Approve/Reject Title Registrations | TC1 – TC11 | Pass | – |
| TS10 | Manage Restricted Words and Affixes | TC1 – TC19 | Pass | – |
| TS11 | Search and View Existing Titles | TC1 – TC15 | Pass | – |
| TS12 | View Submission History | TC1 – TC15 | Pass | – |
| TS13 | Monitor Title Registration History | TC1 – TC8 | Pass | – |
| TS14 | Access Help & Guidelines | TC1 – TC14 | No Run | – |
| TS15 | Contact Support | TC1 – TC13 | Pass | – |
| TS16 | Respond to Support Queries | TC1 – TC12 | Pass | – |

The *Automated Title Verification and Registration System* offers a scalable, intelligent, and user-friendly solution to streamline the title verification process managed by the Press Registrar General of India (PRGI). By leveraging phonetic and semantic similarity detection, enforcing regulatory guidelines, and integrating multilingual awareness, the system ensures that newly submitted titles are both unique and compliant.

With distinct user roles — Title Applicants, PRGI Officials, and System Administrator — the platform provides a well-defined access hierarchy, automated decision-making through similarity scoring, and manual override for special cases. The integrated feedback, history tracking, and support modules enhance transparency and usability.

Extensive use case design and test case coverage have ensured that the system is functionally complete, reliable, and ready for deployment. This project not only solves a real-world administrative bottleneck but also demonstrates a practical application of full-stack development principles, database management, role-based access control, and system validation through comprehensive testing.

Ultimately, this solution fosters fairness, avoids duplication, supports policy enforcement, and sets the stage for future enhancements such as advanced NLP.