Ankit Khushwaha

Education

Indian Institute Of Technology Dharwad

Bachelor of Science, Physics

Dec. 2022 - Present

Dharwad, Karnataka

Technical Skills

Languages: Python, C, HTML, CSS, Bash

Technologies/Frameworks: Flask, Pandas, scikit-learn, SQLAlchemy, Selenium, BeautifulSoup, Jinja, Git, Linux

Tools: REST APIs, Postman, GitHub, GitHub Actions, VS Code, SQLite

Skills: Open Source Development, Machine Learning, Web Scraping, Automation, Dependency & Workflow Management,

Scientific Computing

Experience

Open Source Contributor - Stingray

Feb 2025 - Present

Spectral Timing Analysis Toolkit

Remote

- Added GeneralizedLorentz1D and SmoothBrokenPowerLaw classes to stingray.simulator.models. PR #889
- Added support for reading Fermi/GBM FITS files in EventList.read. PR #894
- Fixed tstart not updating in stingray.Lightcurve.shift. PR #912
- Updated the Stingray documentation interface for better navigation and layout. PR #918

Open Source Contributor - SunPy

 ${\bf Dec}~{\bf 2024-Present}$

Solar Data Analysis Toolkit

Remote

- Added custom energy bin support for Fermi/GBM TimeSeries. PR #7943
- Deprecated sunpy.physics.solar_rotate_coordinate and sunpy.physics.differential_rotate. PR #8039
- Improved the performance of CompositeMap plotting. PR #7948
- Added support for date time.timedelta and astropy.units.Quantity in sunpy.time. TimeRange.shift. PR #7931
- Added support for TimeSeries.truncate to handle out-of-bounds and partial TimeRange inputs. PR #7922
- Added support for the timestamp format %Y%m%d%H%M. PR #7911
- Added release GitHub Action. PR #181
- Updated CI workflow in package-template. PR #184

Projects

Grocerystore Website | Python, Flask, Jinja, Sqlite, SqlAlchemy

()

- Developed a dynamic E-commerce website using the Flask framework, implementing essential features like product listing, cart management, and checkout.
- Implemented role-based authorization with user roles such as Admin and Customer, ensuring appropriate access controls.
- Stored data in a local SQLite database using SQLAlchemy ORM for efficient querying and management.
- Ensured security measures like password hashing (using bcrypt) and secure session management for protecting user credentials and sensitive data.
- Used Jinja templates for dynamic content rendering, improving code maintainability and user interface flexibility.

REST API with Flask | Python, Flask

()

- Developed a RESTful API using Flask to handle HTTP requests.
- Implemented endpoints for creating, reading, updating, and deleting resources, with proper status codes and error handling.
- Used Flask-RESTful extension to simplify the creation of API resources and responses.
- Utilized SQLAlchemy ORM to manage and query data stored in a local SQLite database for efficient data handling.
- Tested API endpoints with Postman to ensure correct functionality and response formatting.