Google Summer of Code 2025

Proposal

Part 1: Biographical Information

Name: Sankalpa Sarkar
Email: sankalpasarkar68@gmail.com
Contact Number: +91-8670707887
University: Adamas University, West Bengal (B.Tech. in Computer Science & Engineering)

LinkedIn: https://www.linkedin.com/in/sankalpacodes/
GitHub: github.com/sanks011
Time Zone: IST (GMT +5:30)

I'm a second-year Computer Science student passionate about open source and accessible software. Over the years, I've contributed to various hackathon projects, focusing on building user-friendly platforms. My current research interests include bridging AI, data visualization, and web development to solve real-world challenges in healthcare, education, and beyond.

Part 2: Personal Experience

I first got involved in open source programming during my freshman year of college. It started as a fascination with how collaborative coding could drive innovation across diverse communities. Since then, I’ve participated in hackathons and built projects like:

- **Lexishift:** A platform providing AI-powered learning tools for individuals with dyslexia.

- **Team Up:** A hackathon teammate-finding application using TypeScript, React, and Firebase.

- **Social Pulse:** A social media analytics dashboard using Flask, React, and LangFlow for AI-driven insights.
- **AdiVote:** A class representative choosing platform via voting using TypeScript, React and Firebase.

These experiences taught me the importance of building solutions that are both robust and easily approachable by non-technical audiences. My love for user-centric development aligns perfectly with the needs of this project: to empower wet-lab scientists by simplifying mass spectrometry tasks.

Part 3: Project Preference and Motivation

Chosen Project: Peptide m/z Calculator

This project resonates with me because it directly addresses a key pain point for researchers: computing the mass-to-charge (m/z) ratio without getting bogged down in code. I believe my background in designing accessible web interfaces (e.g., Lexishift) and my enthusiasm for open source will help me craft a solution that is:

1. **Intuitive and Welcoming:** Wet-lab scientists can simply upload their data, input parameters, and obtain the results in a user-friendly interface.

2. **Technically Sound:** By leveraging pyopenms for calculations, the tool will ensure accurate and efficient results.

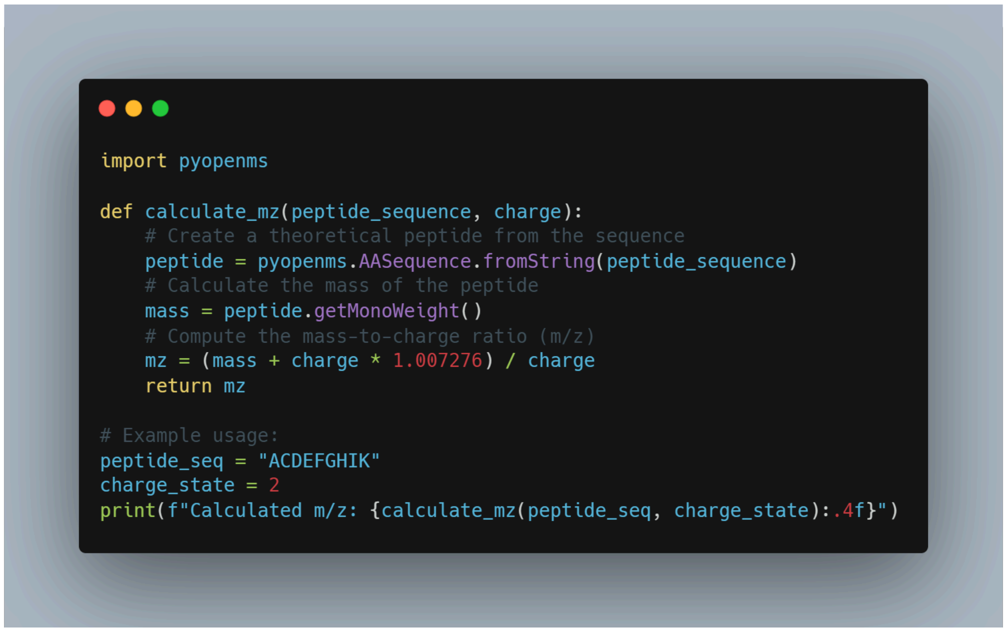
3. **Extendable:** Future expansions could include theoretical spectra generation or additional MS tasks.

I’ve been interested in scientific computing and data analysis, and contributing to OpenMS is an exciting way for me to merge my open-source passion with my desire to help the scientific community.

Part 4: Implementation Plan

I plan to use **Python**, **Streamlit**, and the **OpenMS-WebApps** template to develop a highly interactive, visually guided tool. Below is an overview of my approach:

1. **User Interface & Experience**
   - Develop a **Streamlit** frontend that guides users step-by-step: from uploading peptide sequences to adjusting optional parameters.
   - Include helpful tooltips, error messages, and a tutorial section for those new to MS calculations.
2. **Core Calculation Engine**
   - Integrate **pyopenms** to handle the computation of the mass-to-charge ratio.
   - Ensure input validation (e.g., checking valid amino acid residues, charge states) before sending data to the calculation module.

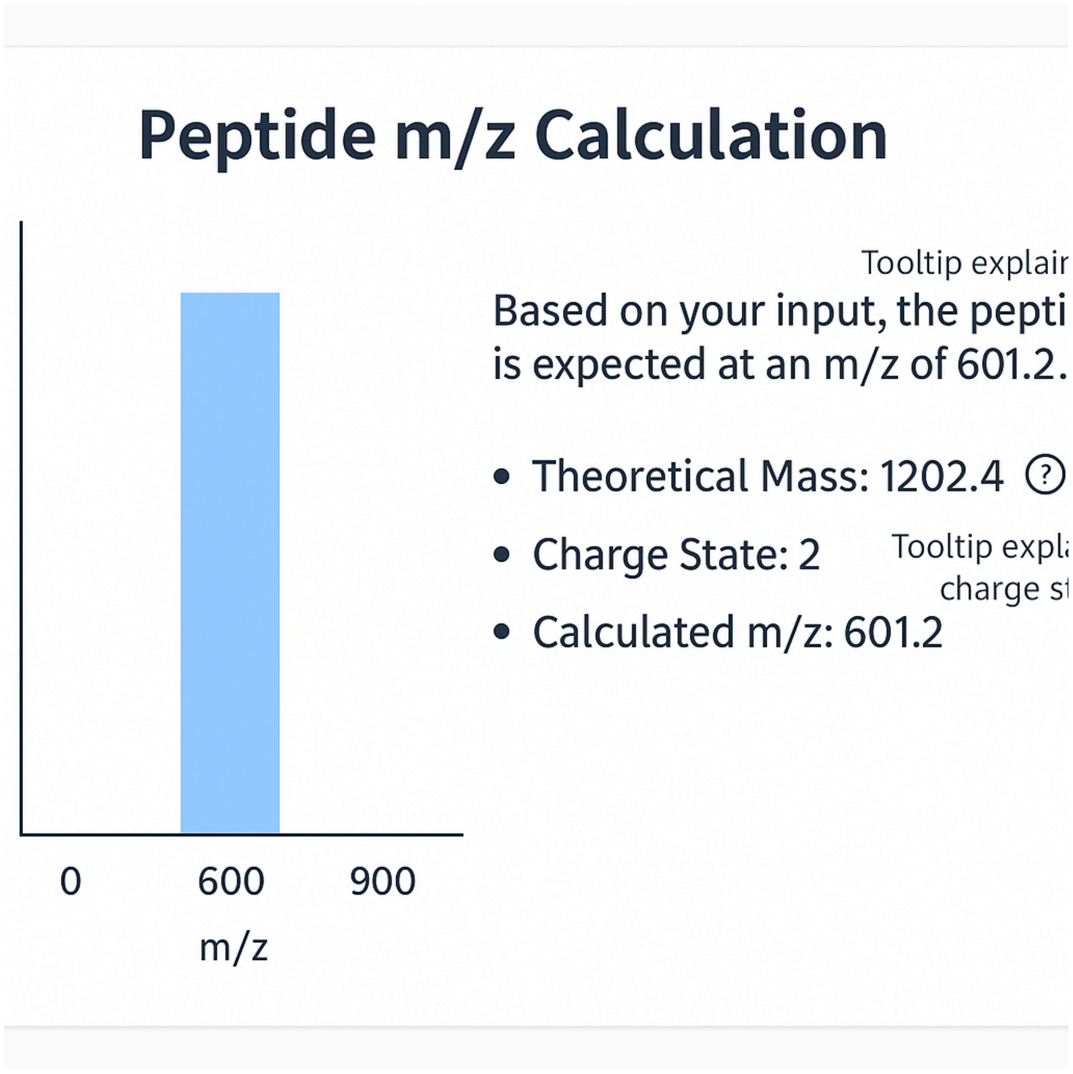


```python
import pyopenms

def calculate_mz(peptide_sequence, charge):
    # Create a theoretical peptide from the sequence
    peptide = pyopenms.AASequence.fromString(peptide_sequence)
    # Calculate the mass of the peptide
    mass = peptide.getMonoWeight()
    # Compute the mass-to-charge ratio (m/z)
    mz = (mass + charge * 1.007276) / charge
    return mz

# Example usage:
peptide_seq = "ACDEFGHIK"
charge_state = 2
print(f"Calculated m/z: {calculate_mz(peptide_seq, charge_state):.4f}")
```

3. **Data Visualization & Interpretation**

    ○ Present computed m/z ratios with interactive charts or highlight potential areas for further analysis.

    ○ Provide a textual summary for quick interpretation (e.g., “Based on your input, the peptide is expected at m/z of X…”).



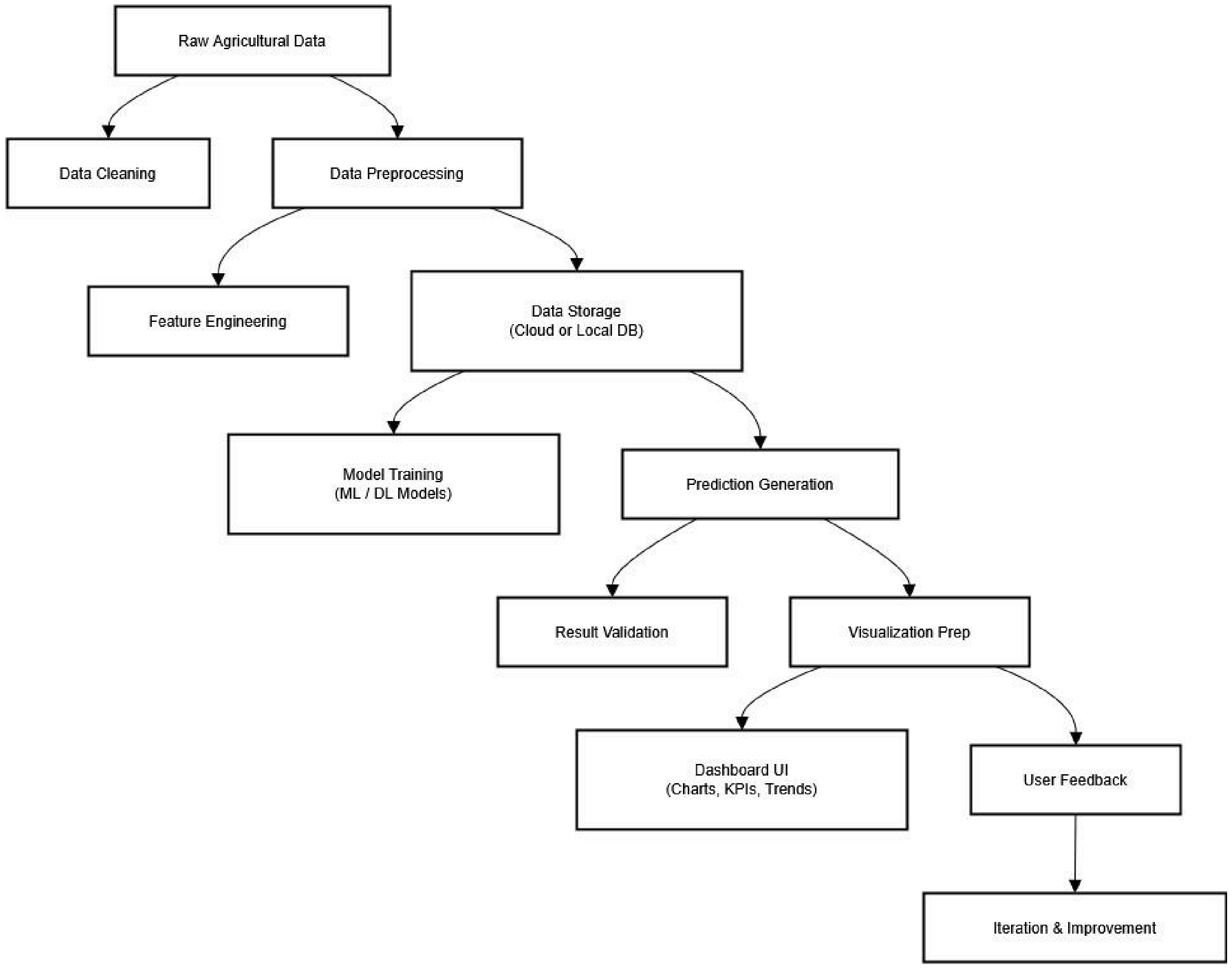
Peptide m/z Calculation

Tooltip explair

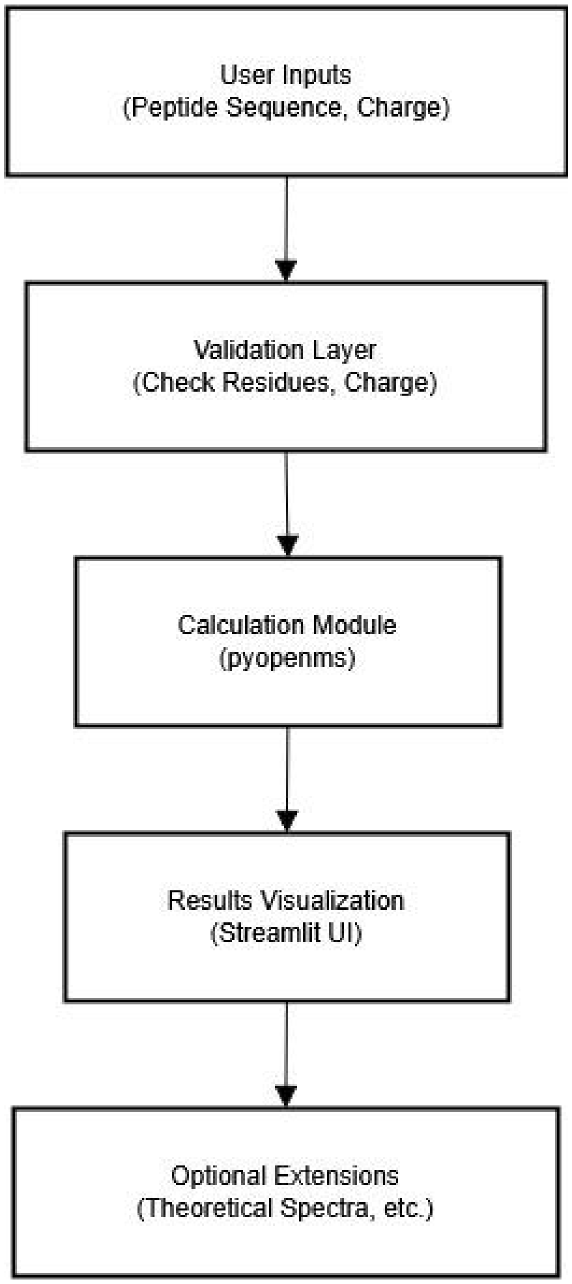
Based on your input, the pepti
is expected at an m/z of 601.2.

- Theoretical Mass: 1202.4 ?
- Charge State: 2
- Calculated m/z: 601.2

Tooltip expla
charge st


Raw Agricultural Data
Data Cleaning
Data Preprocessing
Feature Engineering
Data Storage (Cloud or Local DB)
Model Training (ML / DL Models)
Prediction Generation
Result Validation
Visualization Prep
Dashboard UI (Charts, KPIs, Trends)
User Feedback
Iteration & Improvement

4. **Architecture Diagram**



5. **Testing & Validation**

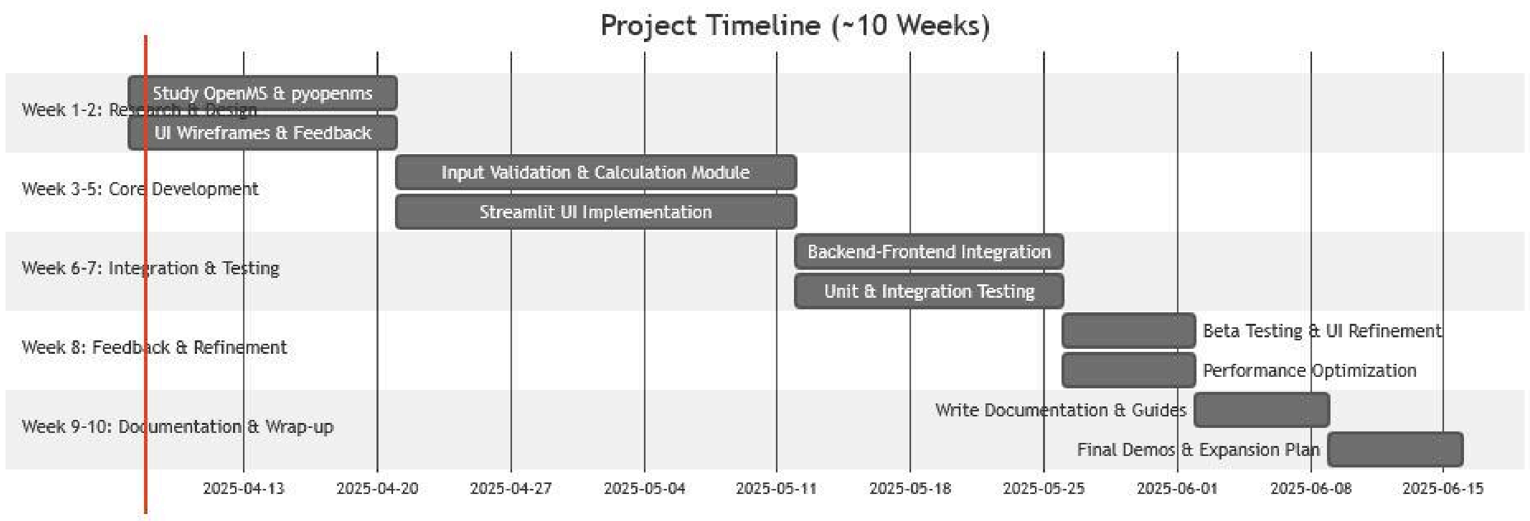
    - **Unit Tests:** Verify that each function (e.g., input validation, calculation) behaves correctly.
    - **Integration Tests:** Confirm that the overall pipeline from user input to final output is robust and handles edge cases (e.g., non-standard amino acids).
    - **User Feedback Loop:** Share a beta version with lab scientists to gather real-world feedback, ensuring that the UI meets their needs.

Part 5: Timeline

Below is a tentative timeline based on a ~10-week project window. I've broken it down into four major phases to ensure smooth progress.

- **Week 1-2: Research & Design**
    - Familiarize myself deeply with OpenMS-WebApps template and pyopenms.
    - Create wireframes for the user interface and gather feedback from mentors and potential end-users.
- **Week 3-5: Core Development**
    - Implement the input validation and calculation modules.
    - Build the initial Streamlit interface for data input and results display.
- **Week 6-7: Integration & Testing**

  - Integrate the backend with the frontend, ensuring a seamless workflow.
  - Develop unit tests for each component and conduct integration tests.
- **Week 8: User Feedback & Refinement**
  - Conduct beta testing with lab scientists and incorporate feedback to refine UI/UX.
  - Optimize performance where needed (e.g., handling large input sets).
- **Week 9-10: Documentation & Finalization**
  - Write comprehensive user documentation, including step-by-step guides and FAQ.
  - Prepare final deliverables, demos, and a plan for potential future expansions



Part 6: Deliverables

1. **Fully Functional m/z Calculator**
   - Web application deployed (e.g., on a public cloud service) allowing easy access to researchers.
2. **Documentation & Tutorials**
   - Clear README and user guide explaining setup, usage, and troubleshooting.
   - Short video or written tutorial for new users.
3. **Test Suite & Reports**
   - Unit tests for core functions (validation, calculation).
   - Integration tests ensuring end-to-end reliability.
   - Performance and user acceptance test reports.
4. **Future Extension Plan**
   - Outline of how to add new MS calculation features (e.g., theoretical spectra generation, advanced QC metrics).
   - Suggestions for integrating with other OpenMS tools or community plugins.

Part 7: Why Choose Me?

1. **Human-Centric Approach:** I’ve consistently focused on making complex technologies approachable, as evidenced by my projects that emphasize accessibility and user experience.

2. **Technical Proficiency:** My background in Python, React, Node.js, and open-source development ensures I can handle both the frontend and backend challenges.

3. **Collaborative Spirit:** Having worked on multiple hackathon teams and open-source projects, I thrive in environments that value mentorship, feedback, and community-driven development.

4. **Enthusiasm for Open Science:** Mass spectrometry is an exciting field, and I’m eager to contribute to OpenMS to help researchers worldwide get quick and accurate insights into their data.

Conclusion:
I’m excited by the prospect of making a tangible impact on the daily workflows of wet-lab scientists. By developing a user-friendly m/z Calculator, I hope to lower the barrier to entry for researchers, encouraging broader participation in computational proteomics. Thank you for considering my proposal - I look forward to the opportunity to collaborate with the OpenMS community!