**Progress Report**

**Week 1:**

We explored the fundamentals of project management and version control using Git and GitHub to ensure smooth collaboration. We formed drafted a detailed project proposal outlining our scope, objectives, and key deliverables. We set up Git repositories, installed Django and React, and setup the project structure.

**Week 2:**

In the second week, we worked on understanding the needs of different user roles, including students, academic registrars, and lecturers. We wrote user stories to capture their specific requirements and documented both functional and non-functional system requirements. We created Django models to represent user roles and primary entities like Issues and Departments. Additionally, we configured React to connect with the Django backend via API endpoints, ensuring smooth frontend-backend communication.

**Week 3:**

During the third week, we designed the system architecture and database schema. We created wireframes and prototypes for the user interfaces, making sure they were intuitive and user-friendly. We developed UML diagrams to visualize system interactions. We submitted an Entity-Relationship Diagram (ERD) and wireframes for the student dashboard and registrar portal. We set up Django REST Framework (DRF) for API development and began implementing React components based on our wireframes.

**Week 4:**

In week four, we worked on implementing authentication and authorization. We designed user roles and permissions to ensure secure access control within the system. We implemented user registration and login using Django and set up React routing for role-based dashboards. We used Django's authentication system to manage user logins and roles while integrating React Router for seamless navigation between user roles.

**Week 5:**

During the fifth week, we focused on developing the student features, particularly the issue submission form. We ensured that issue data was validated and properly stored in the database. We designed a student dashboard to allow students to track the status of their logged issues. We developed and tested the issue submission feature and created a basic dashboard displaying reported issues. We used React to build the issue submission form and set up Django API endpoints for handling issue data.

**Week 6:**

In the sixth week, we worked on developing features for the academic registrar. We implemented functionality for viewing and logged issues. We built an issue management interface using React and created Django API endpoints for filtering and updating issues.

**Problems We Ran Into**

•**Learning Curve:** Some of us were new to tools like Django or React, so setting things up took longer than expected. We had to watch tutorials and troubleshoot errors—like getting the database connected—which ate into our time.

**•Bugs:** We kept getting bugs on both the frontend and backend side that took a lot of trial and error to fix.

**•Time Crunch:** We underestimated how long the issue submission form and dashboards would take. Which left us with a basic version that’s not as polished as we wanted.

**Completed Tasks**

•Allows users log in and see their issues (student or registrar).

•Allows students to submit issues and view their status.

•Allows the registrar view issues.

•Stores everything in the database and tracks changes.

**What’s Next?**

•Allow the registrar to assign issues to lecturers.

•Add features for lecturers to see issues that have been assigned to them and add comments.

•Email verification.

•Start sending notifications (like emails) when issues status.

•Test everything and make it ready for the final hand-in.