Your Flask-based school management application looks solid! It handles user authentication, assignment creation, grading, and roles for students, teachers, and principals. I see that you've implemented key features like:

- User login and registration with role differentiation.
- Assignment creation by students and teachers.
- Assignment grading by teachers, with re-grading by principals.
- Draft assignments for students to edit and submit.
- User comments on assignments.

Some suggestions and improvements you might consider:

1. Validation and Flash Messaging:

For better user experience, ensure flash messages provide clear feedback for all actions. Example: when creating an assignment, adding a confirmation message like "Assignment created successfully!" or when an error occurs during form submission.

2. Submission Verification:

For submit_assignment, ensure that the student cannot submit the assignment after the deadline by adding a check:

```
python
Copy code
if draft_assignment.deadline < datetime.now():
    flash("Cannot submit after the deadline.")
    return redirect(url_for("dashboard"))</pre>
```

3. Input Sanitization and Security:

Make sure all input data is sanitized to prevent SQL injection or XSS attacks. You can use Flask-WTF with CSRF protection enabled to help with form security.

4. Relationship Constraints:

Consider adding cascade deletion for relationships between entities (e.g., when a user is deleted, their profile and assignments should also be deleted). You can achieve this by adding cascade='all, delete-orphan' in relationships:

```
python
Copy code
```

```
profile = db.relationship("Profile", backref="user", uselist=False,
cascade="all, delete-orphan")
```

5. Role-Based Access Control:

For routes like edit_assignment, it's important to ensure proper authorization checks:

```
python
Copy code
@app.route('/edit_assignment/<int:assignment_id>', methods=['GET',
'POST'])
@login_required
def edit_assignment(assignment_id):
    assignment = Assignment.query.get_or_404(assignment_id)
    if assignment.student_id != current_user.id: # Ensure only the
student who created it can edit
    abort(403)
    ...
```

6. Improve Deadline Formatting:

For better handling of the deadline, format it to something more user-friendly when displaying:

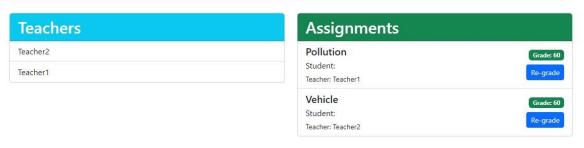
```
python
Copy code
{{ assignment.deadline.strftime('%B %d, %Y, %I:%M %p') }}
7. Images
Principal:
```

Create_assignment:

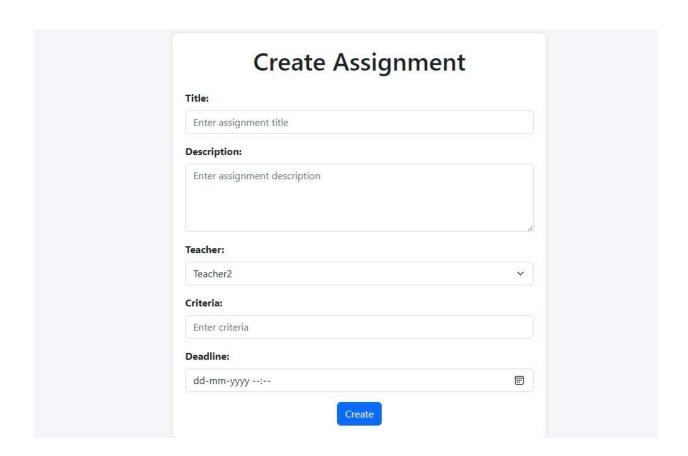
Student: Register: Login: Homepage

Teachers:

Principal Dashboard



Assignments Submitted to Me Vehicle () Grade: 60



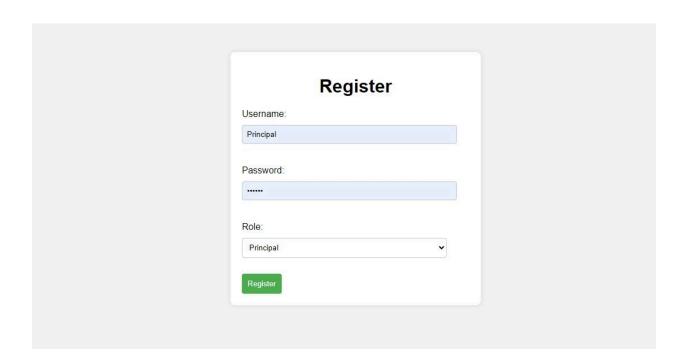
Student Dashboard

Assignments:



Draft Assignments:

Create Assignment



	Login	
Username:		
Principal		
Password:		

