Django Debugging Summary

1. Authentication Credentials Not Provided (401 Unauthorized)

Error: {"detail": "Authentication credentials were not provided."}

Cause: DRF's global DEFAULT_PERMISSION_CLASSES required authentication for all views.

Solution: Set permission_classes = [AllowAny] for RegisterUserView and LoginView.

2. Invalid Credentials Despite Correct Input

Error: {"error": "Invalid credentials"}

Cause: authenticate() returned None because username field was not mapped to phone_number.

Solution:

- Mapped phone_number to username in RegisterUserView.
- Updated CustomUser.save() to ensure phone_number is set as username.

3. Authentication Still Failing

Cause: ModelBackend does not support phone_number for authentication.

Solution: Created a custom authentication backend to authenticate using phone_number.

4. Token Creation Failing

Error: AttributeError: type object 'Token' has no attribute 'objects'

Cause: rest_framework.authtoken was not installed or migrated.

Solution:

- Added 'rest_framework.authtoken' to INSTALLED_APPS.
- Ran migrations to create the Token table.

5. Foreign Key Constraint Failed

Error: django.db.utils.IntegrityError: FOREIGN KEY constraint failed

Cause: Token model was referencing auth.User instead of users.CustomUser.

Solution:

- Set AUTH_USER_MODEL = 'users.CustomUser' in settings.py.
- Reset database and migrations, recreated schema.

6. Everything Works!

Final Outcome: Successfully fixed all issues, and login endpoint returned a token.

Lessons Learned:

1. Always override global permissions for specific endpoints like login.
2. Ensure AUTH_USER_MODEL is set when using custom user models.

3. Use custom authentication backends if needed.

4. Reset migrations and database schema for major model changes.

5. Use debugging logs and database queries to verify data flow.