**Pre-Test Reset**

First, run your reset script:

bash

python reset\_database.py

**Comprehensive Testing Checklist**

**Phase 1: Initial Setup**

* Register as admin user
* Register 3-4 test users (TestUser1, TestUser2, etc.)
* Create Week 1 in admin
* Import Week 1 games via API
* Verify games display correctly with proper spreads
* Check that teams favored >16 points don't appear in pick options

**Phase 2: Week 1 Testing**

* Make picks for 2 users before deadline
* Change one user's pick (test the update feature)
* Leave 1-2 users without picks (for auto-pick testing)
* Check that cumulative spread updates when picks are made
* Verify weekly results page shows "not available" before deadline
* After deadline passes:
  + Run auto-pick process
  + Verify auto-picks selected highest spread teams
  + Check weekly results page now shows all picks
  + Verify AUTO badges appear for auto-picked users
* Mark game results in admin
* Verify lives deducted for wrong picks
* Check cumulative spread calculations are correct
* Verify standings order (lives, then spread tiebreaker)

**Phase 3: Week 2 Testing**

* Create Week 2 and import games
* Verify previously picked teams don't appear in selection
* Make some picks, leave some for auto-pick
* Have one user lose their second life (test elimination)
* Verify eliminated user:
  + Can't make picks
  + Shows as eliminated in standings
  + Appears in eliminated section
* Check My Picks page shows correct history and remaining teams

**Phase 4: Edge Cases to Test**

* **Duplicate registration**: Try registering with same username/email
* **Password change**: Test changing password functionality
* **Pick after deadline**: Try to make/change pick after deadline
* **No eligible teams**: What happens if user has no valid teams <16.5 spread?
* **Untracked teams**: Verify games with non-49 teams display correctly
* **Admin functions**:
  + Reset user password
  + Process auto-picks manually
  + Delete and re-add games

**Phase 5: User Experience Testing**

* **Navigation**: All dropdown menus work
* **Mobile view**: Check responsive design
* **Error messages**: All flash messages appear correctly
* **Data persistence**: Logout/login retains all data
* **Browser back button**: Doesn't cause duplicate submissions

**Testing Scenarios**

**Scenario 1: The Forgetful User**

1. Create a user who never makes picks
2. Verify auto-pick works every week
3. Check they eventually run out of good teams

**Scenario 2: The Strategic Player**

1. Create a user who only picks small favorites/underdogs
2. Verify their cumulative spread stays low
3. Check tiebreaker ranking works correctly

**Scenario 3: The Unlucky Player**

1. Have a user pick losing teams two weeks in a row
2. Verify elimination after losing second life
3. Check they can still view but not participate

**Key Things to Verify**

**Mathematical Correctness**

* **Spread calculation**: -7 favorite adds +7, +3 underdog subtracts 3
* **Auto-pick selection**: Always picks team closest to -16 (most favored)
* **Tiebreaker sorting**: Lives first, then cumulative spread

**Data Integrity**

* **Pick uniqueness**: Can't pick same team twice
* **Week integrity**: Can't have multiple picks per week
* **User state**: Lives and elimination status update correctly

**Security**

* **Authentication**: Can't access pick pages when logged out
* **Authorization**: Only admin can access admin pages
* **Deadline enforcement**: Can't see picks before deadline

**Quick Test Commands**

To speed up testing, you can temporarily modify deadlines in the admin interface or directly in Python:

python

python

>>> from app import app, db

>>> from models import Week

>>> from datetime import datetime, timedelta

>>> with app.app\_context():

... week = Week.query.filter\_by(week\_number=1).first()

... *# Set deadline to 5 minutes from now for testing*

... week.deadline = datetime.utcnow() + timedelta(minutes=5)

... db.session.commit()

**What You've Built Successfully**

You should be proud - you've implemented:

* ✅ Complete user management system
* ✅ Complex pick validation logic
* ✅ Automated pick system with smart selection
* ✅ Comprehensive results tracking
* ✅ Tiebreaker system
* ✅ API integration for game data
* ✅ Admin controls for everything
* ✅ Deadline protection for fairness

The only feature I might suggest adding later is email notifications, but that's not essential for launch.

Run through this testing checklist, and if everything works, you're ready to go live! Let me know if you encounter any issues during testing.