CourtVision Analytics - Project Summary

Project Overview

CourtVision Analytics is a predictive analytics platform for NBA fans that uses historical data and machine learning to forecast game outcomes, visualized through interactive dashboards and accessible via a secure user portal.

Table 1: Foundational Topics (15 pts)

#	Topic	Status	Implementation Details
1	GitHub Repository & Environment Setup	Complete	Public repo with proper .gitignore, virtual environment, Django 4.2+, feature branches
2	Wizard of Oz Prototyping / UI/UX Planning	Complete	Wireframes in /docs/wireframes/ folder (dashboard.png, login.png, core_feature.png, error_state.png)
3	Models + ORM Basics	Complete	Team, Player, Game, UserPick, UserProfile, HistoricalGame, HeadToHead models with migrations and admin registration
4	Views + Templates + URLs	Complete	FBVs and CBVs (ListView, DetailView), template inheritance with base.html, proper URL namespacing
5	User Authentication for Internal Users	Complete	Login/Logout views, @login_required protection, instructor (mohitg2) and guest (infoadmins) accounts
6	Deployment (Production Setup)	Complete	Split settings (base.py, development.py, production.py), PythonAnywhere deployment

Table 2: Functional Add-ons (27 pts = 15 required + 12 bonus)

#	Topic	Status	Implementation Details
1	ORM Queries + Data Summaries	Complete	Aggregations for leaderboards, team stats, prediction accuracy calculations, model ATS/O/U tracking
2	Static Files (CSS/JS Integration)	Complete	Bootstrap 5, custom CSS in /static/css/style.css, proper STATIC_ROOT configuration
3	Charts / Visualization (Matplotlib)	Complete	Team comparison bar charts generated with Matplotlib, embedded as base64 images in game detail pages
4	Forms + Basic Input / CRUD	Complete	UserPickForm for game predictions, ExportForm for data downloads with format and date range options
5	Simple JSON Endpoints / APIs	Complete	Three JSON API endpoints: /api/games/, /api/standings/, /api/teams/{abbr}/ for data sharing and charting
6	Integrate External APIs	Complete	NBA API (balldontlie.io) for game/team data, The Odds API for Vegas betting lines
7	Data Presentation & Export	Complete	CSV and JSON export functionality with predictions, Vegas lines, and user picks
8	User Authentication for External Users	Complete	Public signup with UserCreationForm, auto-login after registration, UserProfile creation
9	External Databases Integration	Complete	MySQL configuration in production.py with environment variables for PythonAnywhere

Total Add-on Topics: 9 (5 required + 4 bonus = +12 bonus points)

Features Summary

Core Features

- **Home Page**: Landing page with featured game of the night, upcoming games preview, conference standings
- **Dashboard**: Today's games with ML predictions, upcoming games, recent results with scores
- **Team Pages**: Team list by conference, team detail with roster, stats, and Four Factors analytics

- **Game Detail**: ML predictions with confidence scores, team comparison charts, community voting
- **Analytics/Leaderboard**: Model accuracy tracking, ATS/O/U performance vs Vegas, top community predictors
- **Data Export**: Download predictions as CSV or JSON with customizable date ranges

Machine Learning Prediction System

- Trained on 7,000+ historical NBA games (2018-2024)
- Features: Elo ratings, Four Factors, home court advantage, rest days, streaks, head-to-head records
- Outputs: Win probability, confidence score, predicted spread, predicted scores

JSON API Endpoints

- `GET /api/games/` - Games data with predictions and Vegas lines (supports ?status= and ?days= filters)
- `GET /api/standings/` - Conference standings with team statistics
- `GET /api/teams/{abbr}/` - Individual team stats and recent game results

Authentication

- Internal instructor account: mohitg2 / graingerlibrary
- Internal guest account: infoadmins / uiucinfo
- Public user registration with favorite team selection

Technical Stack

- Django 4.2+
- Bootstrap 5 (responsive UI)
- Matplotlib (charts)
- scikit-learn (ML predictions)
- MySQL (production) / SQLite (development)

File Structure

```
final_project/
├── courtvision/          # Project settings
│   └── settings/         # Split settings (base, dev, prod)
```

```
├── core/                # Main app
│   ├── models.py        # 7 data models
│   ├── views.py         # View functions + JSON APIs
│   ├── services/        # NBA API + prediction model
│   └── management/      # Custom commands (sync, load, train)
├── accounts/            # Authentication
├── templates/           # HTML templates with inheritance
├── static/              # CSS, JS, images
├── docs/                # Documentation
│   ├── wireframes/      # UI prototypes
│   └── project_summary.md
└── requirements.txt
```

Database Statistics

- 30 NBA Teams
- 90+ Players
- 350+ Games (current season)
- 7,000+ Historical Games (for ML training)
- User picks and profiles tracked

Deployment

- **Platform**: PythonAnywhere
- **URL**: https://justinw2274.pythonanywhere.com
- **Static Files**: Configured with collectstatic
- **Database**: MySQL (external database integration)
- **Scheduled Tasks**: Daily Vegas lines fetch at 3 AM Central

INFO 390 Final Project - University of Illinois