

Python Packages for PEARL Backend

Currently Used (from `pyproject.toml` & `requirements.txt`)

Core Framework & API:

- `fastapi>=0.111.0` - Web framework
- `uvicorn[standard]>=0.30.0` - ASGI server
- `websockets>=12.0` - WebSocket support
- `python-multipart>=0.0.9` - Form data handling

Database & ORM:

- `sqlalchemy>=2.0.30` - Database ORM
- `asyncpg>=0.29.0` - Async PostgreSQL driver
- `alembic>=1.13.0` - Database migrations

Data Validation & Configuration:

- `pydantic>=2.7.0` - Data validation
- `pydantic-settings>=2.3.0` - Settings management
- `python-dotenv>=1.0.0` - Environment variables

Integration:

- `fastapi-mcp>=0.4.0` - MCP integration (custom from GitHub)

Development & Testing:

- `pytest>=8.0.0` - Testing framework
- `pytest-asyncio>=0.23.0` - Async testing
- `pytest-cov>=5.0.0` - Coverage testing
- `httpx>=0.27.0` - HTTP client for testing
- `testcontainers[postgres]>=4.0.0` - Test database containers
- `freezegun>=1.5.0` - Time mocking
- `factory-boy>=3.3.0` - Test data factories
- `mypy>=1.10.0` - Type checking
- `black>=24.0.0` - Code formatting

- `isort>=5.13.0` - Import sorting
- `flake8>=7.0.0` - Linting

Additional Packages Needed for RConnect Deployment

Production Server:

- `gunicorn` - Production WSGI server
- `uvloop` - High-performance event loop

Security & Authentication:

- `python-jose[cryptography]` - JWT handling
- `passlib[bcrypt]` - Password hashing
- `cryptography` - Enhanced cryptographic operations

Monitoring & Logging:

- `loguru` - Enhanced production logging
- `prometheus-client` - Metrics collection
- `sentry-sdk[fastapi]` - Error tracking and monitoring

Performance & Caching:

- `redis` - Caching layer for sessions/data
- `aioredis` - Async Redis client

Data Processing:

- `pandas` - Data manipulation (for reports/exports)
- `openpyxl` - Excel file handling
- `python-docx` - Word document processing

R Packages for PEARL Admin Frontend

Currently Used (from `renv.lock`)

Core Shiny Ecosystem:

- `shiny` (1.11.1) - Core framework
- `bslib` (0.9.0) - Bootstrap 5 theming

- `bsicons` (0.1.2) - Bootstrap icons
- `shinyWidgets` (0.9.0) - Enhanced UI widgets
- `shinyjs` (2.1.0) - JavaScript integration
- `shinyvalidate` (0.1.3) - Form validation
- `shinyTree` (0.3.1) - Tree view components

Data & Tables:

- `DT` (0.33) - Interactive data tables
- `dplyr` (1.1.4) - Data manipulation
- `lubridate` (1.9.4) - Date/time handling
- `readxl` (1.4.5) - Excel file reading
- `openxlsx` (4.2.8) - Excel file writing

HTTP & API Communication:

- `httr` (1.4.7) - HTTP client (legacy)
- `httr2` (1.2.1) - Modern HTTP client
- `jsonlite` (1.8.8) - JSON processing
- `curl` (6.4.0) - HTTP backend

Real-time & WebSockets:

- `websocket` (1.4.4) - WebSocket client
- `later` (1.3.2) - Async scheduling
- `promises` (1.3.3) - Async promises

UI & Styling:

- `htmltools` (0.5.8.1) - HTML utilities
- `htmlwidgets` (1.6.4) - JavaScript widgets
- `fontawesome` (0.5.2) - Font Awesome icons
- `sass` (0.4.9) - CSS preprocessing

Configuration & Environment:

- `dotenv` (1.0.3) - Environment variables
- `renv` (1.0.7) - Package management

Supporting Libraries:

- `R6` (2.5.1) - Object-oriented programming
- `rlang` (1.1.6) - Language utilities
- `cli` (3.6.2) - Command line interface
- `glue` (1.7.0) - String interpolation
- `magrittr` (2.0.3) - Pipe operators
- `digest` (0.6.35) - Cryptographic hashing

Additional Packages Needed for RConnect Deployment

Essential RConnect Integration:

- `rsconnect` - Deploy and manage applications on RStudio Connect
- `config` - Environment-specific configuration management (replace dotenv)
- `rstudioapi` - Access RConnect user information and session details

Enhanced Logging & Monitoring:

- `logger` - Structured application logging for production
- `futile.logger` - Alternative robust logging framework (optional)
- `tryCatchLog` - Enhanced error logging and debugging

Authentication & Security:

- `jose` - JWT token handling for API authentication

Enhanced UI/UX for Production:

- `shinycssloaders` - Loading indicators and spinners
- `waiter` - Loading screens and progress bars
- `shinyfeedback` - User feedback and notification components
- `shinyalert` - User notifications

Performance & Optimization:

- `memoise` (2.0.1) - Already have - function memoization
- `cachem` (1.1.0) - Already have - caching utilities
- `pool` - Database connection pooling (if needed)
- `future` - Async processing (if needed)

