ORBITS - PLAN

Carlijn van der Vleuten
Panagiotis Evangelou
Ron L. Tabuchov
Bartosz Kudyba
Niels Meijer

Data Science and Artificial Intelligence, Breda University of Applied Science

Index

1 Project Overview

ORBITS is a platform that recommends suitable workplaces and job opportunities to candidates based on their CVs by leveraging different AI techniques.

2 Project Goal

Develop an end-to-end solution that ingests candidate CVs, processes and analyse their content, and generates ranked job and company recommendations tailored to each candidate’s profile.

3 Method

We will follow an **Agile Scrum** approach organized into five sprints (Sprint 0–4). Daily stand-ups at **09:30** will ensure alignment and rapid feedback. In addition, group meeting with mentor will provide insight when obstacles occurs.

4 Roles

- **Carlijn van der Vleuten:** Data responsible
- **Panagiotis Evangelou:** Front-end development
- **Ron L. Tabuchov:** Scrum Master & documentation, overall coordination
- **Bartosz Kudyba:** Modelling & algorithm design
- **Niels Meijer:** Pipeline, security & integration

5 Sprint Breakdown

Sprint	Dates	Focus & Deliverables
0	7 May – 12 May	Planning: project roadmap, environment & data foundation
1	13 May – 23 May	Python package foundation, testing & documentation
2	26 May – 6 Jun	Data pipeline setup, Azure ML environments & model training
3	9 Jun – 16 Jun	API and GUI development, integration and review
4	16 Jun – 27 Jun	Final testing, QA, presentations and project handover

6 Task Breakdown

Sprint 0 (7 May – 12 May)

Goal: Finish all planning, environment setup & data foundations.

- Project Plan & Roadmap
- Setup + Installation
  - Install Tailscale
  - Install Postgres
  - Set up Figma
  - Set up Trello
- Resume dummy data
  - Set up PRAW (Reddit API)
  - Retrieve image URLs
  - OCR images into text
  - Write extracted text into the database
- Database facilitating
  - Provision dev database + user roles
  - Provision vector database + user roles
  - Move data snapshot into dev database
  - Fix None/empty lists in jobs table

- Python Package
    - Create modular package installable via pip/whl
    - Implement fully-functional CLI
    - Expose web-interface/API
    - Write comprehensive docs
    - Containerize for Azure & local deployment
    - Create demo of functionality & MLOps best practices
- Python Package Planning 1-4: Helper, Process, inference, Ingestion

Sprint 1 (13 May – 23 May)

Goal: Deploy first version and document process.

- Unit Testing
- Logging
- CLI refinement
- Docstrings + poetry
- API Cloud Deployment S1
- GUI for user interaction with API
- Deployment demo 1#
- API Development
- Base Front End development

Tasks:

- ReactJS + Vite website foundation
- Website file structure organization
- Landing page
- Mobile scaling
- Job list section
- JavaScript API connectivity for web buttons
- Blob storage
- FileUpload
- CI/CD Python Packages
- Flake8 automation & test
- Coverage report
- DRF to Fast API communication
- Helper package
- Log manager
- Database operation
- Inference package
- Functions
- Hard filter query
- Vectorizer
- Feature extractor
- Registration, login, logout endpoint

Sprint 2 (24 May – 6 June)

Goal: Full Basic Functionality Deployed of the website (Log in, CV Upload, Jobs, Filters, Scores)

- Setup F-flow data pipeline
- Coverage report setup 85%
- Export documentations to GitHub
- Testing CV's
- User prompt search
- Cloud
- Sphinx for ML microservice

- Data Mgmt Azure ML
  - Upload data to an Azure ML datastore
  - Create versioned data assets (train/test/validation)
  - Create & register Azure ML environments for train/eval/inference
  - Test those environments with a small-scale training job
- Azure ML data pipeline & model training
  - Ingest data from the versioned assets
  - Perform data preprocessing
  - Train the model
  - Evaluate model performance
  - Register the model if criteria are met
- Model Documentation
- MLOps
- Model Monitoring

Sprint 3 (7 June – 16 June)

Goal: Deploy, review and test the API & UI.

- Catch up with unaccomplished tasks
- Deploy on portainer and cloud:
  - Model ml azure
  - Backend ml apps
  - Database– container with backend or hosted in cloud
- Testing
  - Test the application on a range of scenarios to ensure accuracy, reliability & scalability
  - Validate the application's performance using relevant evaluation metrics
  - Ensure the application is robust to common errors and exceptions
- User Tests
- Documentation
- Automation
- Redirect between two models
- Gathering user feedback
- Review ILO's

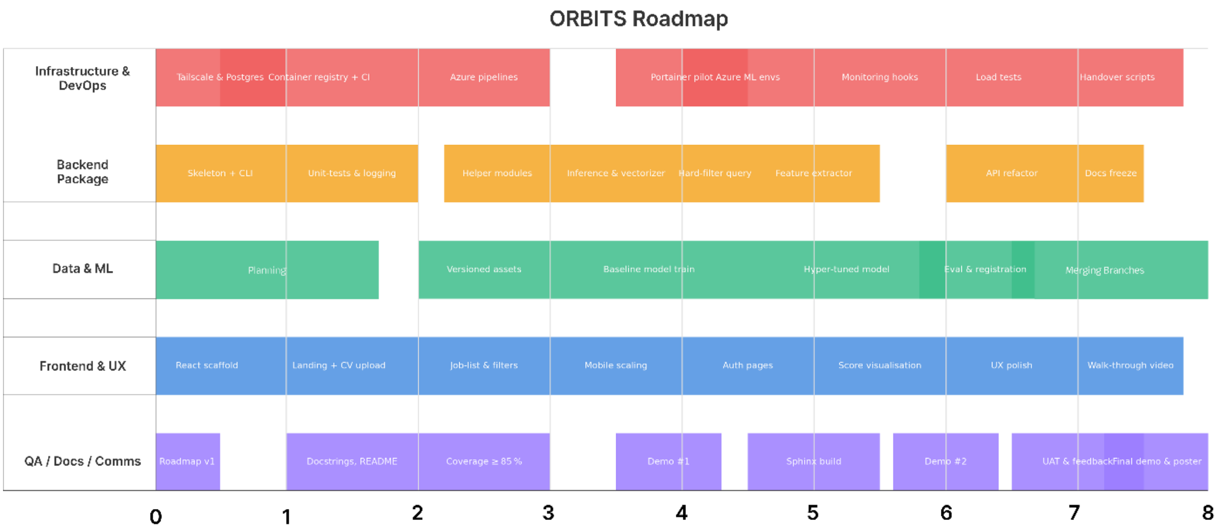
Sprint 4 (16 June – 27 June)

Goal: Final testing, refinement, QA, dashboard, presentations and evidencing.

7 Risks

Risk	Mitigation Strategy
Team member sickness or unavailability	Cross-training, backlog buffer, flexible resourcing
Personal or academic workload conflicts	Proactive sprint planning, task reallocation
Server or connectivity issues	Backup environments, offline testing, cloud redundancy
Data quality or format inconsistencies	Validation scripts, early data profiling

8 Roadmap



9 Tools & Governance

Tools & Governance

- **Version Control**: GitHub
- **Work Management**: Azure DevOps (Epics, Issues, Tasks, Iterations)
- **CI/CD & MLOps**: Azure Pipelines & Azure ML