



TITANIC - 7UP

PRESENTATION

OVERVIEW - TITANIC AI

Technologies Used:

- React
- FastAPI
- Docker
- Scikit-learn
- GitLab
- CI/CD
- JWT(for Auth)
- Postgres
- SQL Alchemy
- Python
- Uvicorn

Group Members:

- Jamal Dassrath 22301035
- Erbakan Ahmad 12306435
- Love 12306406
- Muhammad Ahtisham Bhatti 22301502
- Sonali Elabada Arachchige 22301094
- Utkarshni 22306093
- Marlis Wagner 22301867





AGILE & SPRINT

Scrum methodology – 3 Sprints

Tasks managed using **GitLab Issue Board**

Labels: **Open → In Progress → Blocked → Closed**

Weekly Standups: **+ 3 sprint reviews**



SYSTEM ARCHITECTURE

- Web Frontend (SPA with React)
- Web Backend (FastAPI + JWT + SQLAlchemy)
- Model Backend (FastAPI + Scikit-learn)
- PostgreSQL Database
- All connected through Docker Compose and reverse proxy (Caddy).



FRONTEND OVERVIEW

REACT SPA WITH RESPONSIVE DESIGN
USER INPUT SLIDERS & MODEL SELECTION
REAL-TIME PREDICTIONS
E2E TESTING WITH CYPRESS*
HISTORY DISPLAY*

MAINLY BY UTKARSHNI AND AHTISHAM




*** = STILL TESTING**



WEB BACKEND

FASTAPI-BASED REST API
USER REGISTRATION, LOGIN, TOKEN AUTH
PREDICTION HISTORY (LAST 10 PER USER)*
INTEGRATION WITH MODEL BACKEND
PYTEST TESTS IN CI*



MAINLY BY SONALI AND ERBAKAN

*** = STILL TESTING**



MODEL BACKEND

FASTAPI ML SERVICE

MODELS: RANDOM FOREST, SVM, KNN, ETC.

TRAINED MODELS PERSISTED WITH PICKLE

ADMIN TOOLS FOR NEW MODEL TRAINING*

INTERNAL API ONLY (DOCKER NETWORK)

MAINLY BY – MARLIS AND LOVE

*** = STILL TESTING**



DEVOPS & DOCKERIZATION

DOCKERIZED ALL SERVICES
GITLAB CI/CD FOR BUILDS AND TESTS*
COMPOSE.YAML FOR ORCHESTRATION
CADDY REVERSE PROXY CONFIG

JAMAL



* = *STILL TESTING*



SCRUM/SPRINTS


Sprint 1

- Core Structure Setup
- Task Allocation
- Pipeline Template
- Meeting Scheduling

Sprint 2

- Auth system
- API integration
- Frontend logic
- Model Integration
- Docker Integration

Sprint 3

- Admin Pages
 - Persistent storage
 - Testing
 - Docker Refinement
 - Bug Fixes
 - Wrapping Up
- 

Survival calculator
with multiple models

Auth system:
login with
persistent history



Realtime
predictions with
UI feedback

Full Docker
Containerization

**KEY
FEATURES**




PROBLEMS AND SOLUTIONS



| Problems Encountered | Solutions |
|---|---------------------------------------|
| Merge Conflicts | Merge Requests with Requests Approval |
| Lost Teammates | Emergency Discord Meetings |
| Teammates Disagreements | Vote or Polls made |
| Unclear feature requirements | Writing it down |
| Lack of testing leads to last-minute bugs | Group Code Review |



LESSONS LEARNT

- *GIT ETIQUETTE*
 - *WORKING IN A TEAM IS TOUGH*
 - ***ALWAYS PULL BEFORE PUSHING***
 - SCRUM METHODOLOGIES
 - CONTAINERIZATION
 - **COMMUNICATION**
 - *WORKING WITH POLYREPO ARCHITECTURE*
- 
- 