## Task Dependencies

Preliminary Work Preliminary Work PV2. Architectural Design PW1 Preliminary Work PV3. Technical Documentation PV2 Machine Learning & Al Module - Data Processing PV3 Machine Learning & Al Module - Data Processing PV3 Machine Learning & Al Module - Data Processing PV3 Machine Learning & Al Module - Model Training PV1. Train deep learning models PV2 Machine Learning & Al Module - Model Training PV1. Train deep learning models PV2 Machine Learning & Al Module - Model Training PV1. Train deep learning models PV2 Machine Learning & Al Module - Model Evaluation ME1. Evaluate model performance MT2 Machine Learning & Al Module - Model Evaluation ME2. Benchmark Al models ME1 Machine Learning & Al Module - API Integration and Resilience API1. Integrate GEMINI API PV3 Machine Learning & Al Module - API Integration and Resilience API2. Test fallback mechanisms API1 Web - Backend BE1. Set up backend architecture PV3 Web - Backend BE2. Implement user authentication BE1 Web - Backend BE3. Develop APIs for image upload BE2 Web - Backend BE3. Develop APIs for image upload BE2 Web - Backend BE5. Enable real-time data updates BE3 Web - Backend BE6. Secure APIs with tokens BE5 Web - Frontend FE1. Create login and sign-up UI PV3 Web - Frontend FE2. Develop Home Page FE1 Web - Frontend FE3. Build disease analysis components FE2 Web - Frontend FE4. Design interactive map FE3 Web - Frontend FE5. Implement community page UI FE4 Web - Frontend FE6. Test accessibility FE5	PM: Ayşe PM: Ayşe PM: Ayşe AI Dev: Irmak AI Dev: Irmak AI Dev: Irmak AI Dev: Irmak QA: Pamir QA: Pamir DevOps: Emre QA: Pamir
Preliminary Work  Machine Learning & Al Module - Data Processing  DP1. Preprocess datasets  PW3  Machine Learning & Al Module - Data Processing  DP2. Implement image processing pipeline  Machine Learning & Al Module - Model Training  MT1. Train deep learning models  DP2  Machine Learning & Al Module - Model Training  MT2. Develop disease detection algorithms  MT1  Machine Learning & Al Module - Model Evaluation  ME1. Evaluate model performance  MT2  Machine Learning & Al Module - Model Evaluation  ME2. Benchmark Al models  ME1  Machine Learning & Al Module - API Integration and Resilience  API1. Integrate GEMINI API  PW3  Machine Learning & Al Module - API Integration and Resilience  API2. Test fallback mechanisms  API1  Web - Backend  BE1. Set up backend architecture  PW3  Web - Backend  BE2. Implement user authentication  BE1  Web - Backend  BE3. Develop APIs for image upload  BE2  Web - Backend  BE4. Configure database  BE3  Web - Backend  BE5. Enable real-time data updates  BE4  Web - Backend  BE5. Enable real-time data updates  BE4  Web - Backend  FE6. Secure APIs with tokens  BE5  Web - Frontend  FE1. Create login and sign-up UI  PW3  Web - Frontend  FE2. Develop Home Page  FE1  Web - Frontend  FE3. Build disease analysis components  FE2  Web - Frontend  FE4. Design interactive map  FE3  Web - Frontend  FE5. Implement community page UI  FE4  Web - Frontend	PM: Ayşe Al Dev: Irmak Al Dev: Irmak Al Dev: Irmak Al Dev: Irmak QA: Pamir QA: Pamir DevOps: Emre QA: Pamir TTL: İrem
Machine Learning & Al Module - Data Processing Machine Learning & Al Module - Data Processing Machine Learning & Al Module - Model Training Machine Learning & Al Module - Model Training MT1. Train deep learning models DP2 Machine Learning & Al Module - Model Training MT2. Develop disease detection algorithms MT1 Machine Learning & Al Module - Model Evaluation ME1. Evaluate model performance MT2 Machine Learning & Al Module - Model Evaluation ME2. Benchmark Al models ME1 Machine Learning & Al Module - API Integration and Resilience MP1. Integrate GEMINI API Machine Learning & Al Module - API Integration and Resilience API2. Test fallback mechanisms API1 Meb - Backend BE1. Set up backend architecture PW3 Web - Backend BE2. Implement user authentication BE1 Web - Backend BE3. Develop APIs for image upload BE2 Web - Backend BE4. Configure database BE5. Enable real-time data updates BE4 Web - Backend BE5. Secure APIs with tokens BE5 Web - Frontend FE1. Create login and sign-up UI PW3 Web - Frontend FE2. Develop Home Page FE3 Web - Frontend FE4. Design interactive map FE3 Web - Frontend FE5. Implement user autheromanity page UI FE4 Web - Frontend FE5. Implement user autheromanity page UI FE4 Web - Frontend FE5. Implement user autheromanity page UI FE4 Web - Frontend FE5. Implement user autheromanity page UI FE4 Web - Frontend FE5. Implement community page UI FE5	Al Dev: Irmak QA: Pamir QA: Pamir DevOps: Emre QA: Pamir TTL: İrem
Machine Learning & Al Module - Data Processing  DP2. Implement image processing pipeline  DP1  Machine Learning & Al Module - Model Training  MT1. Train deep learning models  DP2  Machine Learning & Al Module - Model Training  MT2. Develop disease detection algorithms  MT1  Machine Learning & Al Module - Model Evaluation  ME1. Evaluate model performance  MT2  Machine Learning & Al Module - Model Evaluation  ME2. Benchmark Al models  ME1  Machine Learning & Al Module - API Integration and Resilience  API1. Integrate GEMINI API  Machine Learning & Al Module - API Integration and Resilience  API2. Test fallback mechanisms  API1  Web - Backend  BE1. Set up backend architecture  PW3  Web - Backend  BE2. Implement user authentication  BE1  Web - Backend  BE3. Develop APIs for image upload  BE2  Web - Backend  BE4. Configure database  BE3  Web - Backend  BE6. Secure APIs with tokens  BE6. Secure APIs with tokens  BE6.  Web - Frontend  FE1. Create login and sign-up UI  PW3  Web - Frontend  FE2. Develop Home Page  FE3  Web - Frontend  FE3. Build disease analysis components  FE3  Web - Frontend  FE5. Implement community page UI  FE4  Web - Frontend  FE6. Test accessibility  FE5	Al Dev: Irmak Al Dev: Irmak Al Dev: Irmak QA: Pamir QA: Pamir DevOps: Emre QA: Pamir TTL: İrem
Machine Learning & Al Module - Model Training  MT1. Train deep learning models  MT2. Develop disease detection algorithms  MT1  Machine Learning & Al Module - Model Evaluation  ME1. Evaluate model performance  MT2  Machine Learning & Al Module - Model Evaluation  ME2. Benchmark Al models  ME1  Machine Learning & Al Module - API Integration and Resilience  API1. Integrate GEMINI API  Machine Learning & Al Module - API Integration and Resilience  API2. Test fallback mechanisms  API1  Web - Backend  BE1. Set up backend architecture  PW3  Web - Backend  BE2. Implement user authentication  BE1  Web - Backend  BE3. Develop APIs for image upload  BE2  Web - Backend  BE4. Configure database  BE4  Web - Backend  BE5. Enable real-time data updates  BE4  Web - Backend  BE6. Secure APIs with tokens  BE6  Web - Frontend  FE1. Create login and sign-up UI  PW3  Web - Frontend  FE2. Develop Home Page  FE1  Web - Frontend  FE3. Build disease analysis components  FE2  Web - Frontend  FE5. Implement community page UI  FE4  Web - Frontend  FE5. Implement community page UI  FE4  Web - Frontend  FE6. Test accessibility  FE5	Al Dev: Irmak Al Dev: Irmak QA: Pamir QA: Pamir DevOps: Emre QA: Pamir TTL: İrem
Machine Learning & Al Module - Model TrainingMT2. Develop disease detection algorithmsMT1Machine Learning & Al Module - Model EvaluationME1. Evaluate model performanceMT2Machine Learning & Al Module - Model EvaluationME2. Benchmark Al modelsME1Machine Learning & Al Module - API Integration and ResilienceAPI1. Integrate GEMINI APIPW3Machine Learning & Al Module - API Integration and ResilienceAPI2. Test fallback mechanismsAPI1Web - BackendBE1. Set up backend architecturePW3Web - BackendBE2. Implement user authenticationBE1Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	Al Dev: Irmak  QA: Pamir  QA: Pamir  DevOps: Emre  QA: Pamir  TTL: İrem
Machine Learning & Al Module - Model EvaluationME1. Evaluate model performanceMT2Machine Learning & Al Module - Model EvaluationME2. Benchmark Al modelsME1Machine Learning & Al Module - API Integration and ResilienceAPI1. Integrate GEMINI APIPW3Machine Learning & Al Module - API Integration and ResilienceAPI2. Test fallback mechanismsAPI1Web - BackendBE1. Set up backend architecturePW3Web - BackendBE2. Implement user authenticationBE1Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	QA: Pamir QA: Pamir DevOps: Emre QA: Pamir TTL: İrem
Machine Learning & Al Module - Model EvaluationME2. Benchmark Al modelsME1Machine Learning & Al Module - API Integration and ResilienceAPI1. Integrate GEMINI APIPW3Machine Learning & Al Module - API Integration and ResilienceAPI2. Test fallback mechanismsAPI1Web - BackendBE1. Set up backend architecturePW3Web - BackendBE2. Implement user authenticationBE1Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	QA: Pamir DevOps: Emre QA: Pamir TTL: İrem
Machine Learning & Al Module - API Integration and ResilienceAPI1. Integrate GEMINI APIPW3Machine Learning & Al Module - API Integration and ResilienceAPI2. Test fallback mechanismsAPI1Web - BackendBE1. Set up backend architecturePW3Web - BackendBE2. Implement user authenticationBE1Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	DevOps: Emre  QA: Pamir  TTL: İrem
Machine Learning & Al Module - API Integration and ResilienceAPI2. Test fallback mechanismsAPI1Web - BackendBE1. Set up backend architecturePW3Web - BackendBE2. Implement user authenticationBE1Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	QA: Pamir TTL: İrem
Web - BackendBE1. Set up backend architecturePW3Web - BackendBE2. Implement user authenticationBE1Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	TTL: İrem
Web - BackendBE2. Implement user authenticationBE1Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	
Web - BackendBE3. Develop APIs for image uploadBE2Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	TTI . İvana
Web - BackendBE4. Configure databaseBE3Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	TTL: İrem
Web - BackendBE5. Enable real-time data updatesBE4Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	TTL: İrem
Web - BackendBE6. Secure APIs with tokensBE5Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	TTL: İrem
Web - FrontendFE1. Create login and sign-up UIPW3Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	TTL: İrem
Web - FrontendFE2. Develop Home PageFE1Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	TTL: İrem
Web - FrontendFE3. Build disease analysis componentsFE2Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	Mobile Dev: Efe
Web - FrontendFE4. Design interactive mapFE3Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	Mobile Dev: Efe
Web - FrontendFE5. Implement community page UIFE4Web - FrontendFE6. Test accessibilityFE5	Mobile Dev: Efe
Web - Frontend FE6. Test accessibility FE5	Mobile Dev: Efe
,	Mobile Dev: Efe
Non Doy Tooks	Mobile Dev: Efe
INUT-DEV Tasks	PM: Ayşe
Non-Dev Tasks ND2. TL	TTL: İrem
Non-Dev Tasks - Quality Assurance (QA) QA1. Test Planning PW3	QA: Pamir
Non-Dev Tasks - Quality Assurance (QA) QA2. Functional Testing QA1	QA: Pamir
Non-Dev Tasks - Quality Assurance (QA) QA3. Regression Testing QA2	QA: Pamir
Non-Dev Tasks - Quality Assurance (QA) QA4. Performance Testing QA3	QA: Pamir
Non-Dev Tasks - Quality Assurance (QA) QA5. Security Testing QA4	QA: Pamir
Non-Dev Tasks - Quality Assurance (QA) QA6. Conduct Smoke Tests QA5	QA: Pamir
Non-Dev Tasks - Bug-fix BF1. Identify and resolve bugs QA1, QA2	QA: Pamir
Non-Dev Tasks - Bug-fix BF2. Verify bug fixes QA3, QA4	QA: Pamir
DevOps DO1. Staging Deployment QA5, QA6	
DevOps DO2. Production Deployment BF1	DevOps: Emre