

Final Project Guidelines

Scope Definition

- Clear problem statement
- Clinical relevance
- Realistic objectives
- Innovation component

Team Formation

- 2-4 members recommended
- Complementary skills
- Clear role division
- Individual contribution tracking

Timeline Milestones

- Week 1: Proposal submission
- Week 3: Progress check-in
- Week 5: Draft results
- Week 7: Final presentation

Deliverables

- Code repository (GitHub)
- Technical report (10-15 pages)
- Presentation slides
- Demo video (5-10 min)

Evaluation Rubric: Technical merit (30%), Innovation (25%), Clinical relevance (20%), Presentation (15%), Documentation (10%)

1

Scope Definition: Building a Strong Foundation

A well-defined project scope is the cornerstone of success. It ensures your team has a clear direction, measurable goals, and a focused approach to solving a meaningful problem in medical AI.



Clear Problem Statement

Define what specific medical challenge you're addressing. Be precise about the clinical scenario, patient population, and expected outcomes.



Clinical Relevance

Ensure your project addresses a real clinical need. Research current practices, identify gaps, and demonstrate how your solution improves patient care.



Realistic Objectives

Set achievable goals within the project timeframe. Break down complex problems into manageable milestones with measurable success criteria.



Innovation Component

Incorporate novel approaches or improvements to existing methods. Consider unique data sources, algorithms, or clinical workflows.



Example Project Scope

Problem: Early detection of diabetic retinopathy in rural areas with limited ophthalmologist access.

Clinical Relevance: 30% of diabetic patients in rural regions lack regular eye screening, leading to preventable blindness.

Objectives: Develop a mobile app using deep learning to classify retinal images with 90%+ accuracy, deployable on smartphones.

Innovation: Integration of lightweight CNN architecture optimized for low-resource settings with offline functionality.



Pro Tips

- Start with literature review to understand current state-of-the-art
- Consult with medical professionals to validate clinical relevance
- Use SMART criteria (Specific, Measurable, Achievable, Relevant, Time-bound) for objectives
- Document all assumptions and constraints early

2

Team Formation: Assembling Your Dream Team

Effective teamwork is essential for project success. The right combination of skills, clear communication, and well-defined roles ensure smooth collaboration and high-quality outcomes.



Team Size: 2-4 Members

2 members: Faster decision-making, closer collaboration

3 members: Balanced workload, diverse perspectives

4 members: Maximum skill diversity, specialized roles



Complementary Skills

- Machine Learning Engineer
- Clinical/Medical Expert
- Data Scientist/Analyst
- Software Developer
- Project Manager



Role Division

Data Lead: Dataset curation, preprocessing

Model Lead: Architecture design, training

Clinical Lead: Medical validation, relevance

Integration Lead: System deployment, testing



Contribution Tracking

- Use GitHub commits and branches
 - Weekly progress reports
- Individual responsibility matrix
- Peer evaluation system



Sample Team Structure

3-Person Team for Pneumonia Detection Project:

Member 1 (ML Engineer): Designs and trains CNN model, hyperparameter tuning, model optimization

Member 2 (Medical Student): Validates clinical relevance, reviews radiological annotations, prepares medical context

Member 3 (Data Scientist): Data preprocessing, augmentation strategies, statistical analysis, visualization of results



Pro Tips

- Hold a kickoff meeting to align expectations and set ground rules
- Use project management tools (Trello, Asana, Notion) for task tracking
- Schedule regular sync meetings (at least weekly)
- Establish clear communication channels (Slack, Discord, etc.)
- Document all decisions and meeting notes in shared repository

3

Timeline Milestones: Your Roadmap to Success

A structured timeline with clear milestones keeps your project on track and ensures steady progress. Each checkpoint serves as an opportunity to assess, adjust, and improve your approach.

W1

Week 1: Proposal Submission

Submit a comprehensive project proposal including problem statement, methodology, dataset description, expected outcomes, and team member roles. Include preliminary literature review and feasibility analysis.

W3

Week 3: Progress Check-in

Present initial data exploration results, baseline model performance, and preliminary findings. Discuss challenges encountered and proposed solutions. Receive feedback from instructors and adjust approach if needed.

W5

Week 5: Draft Results

Submit draft technical report with complete methodology, experimental results, initial analysis, and visualizations. Model should be fully trained with performance metrics documented. Begin preparing presentation materials.

W7

Week 7: Final Presentation

Deliver final presentation showcasing complete project, including demo video, comprehensive results, clinical implications, and lessons learned. Submit all deliverables: code, report, slides, and demo.



Detailed Week-by-Week Breakdown

Week 1-2: Dataset acquisition, exploratory data analysis, baseline model setup

Week 3-4: Model architecture design, initial training, hyperparameter tuning

Week 5-6: Advanced experiments, performance optimization, result analysis, report drafting

Week 7: Final polishing, presentation preparation, demo video creation, documentation finalization



Pro Tips

- Build in buffer time for unexpected challenges (model convergence issues, data problems)
- Start writing documentation early, not just at the end
- Prepare backup plans for each milestone in case of technical setbacks
- Use version control (Git) to track progress and enable rollback if needed
- Don't wait until Week 7 to test your demo—test continuously throughout

High-quality deliverables demonstrate the depth and rigor of your work. Each component serves a specific purpose in communicating your project's value to different audiences.



Code Repository (GitHub)

Structure: Well-organized folders (data/, models/, src/, notebooks/, docs/)

Contents: All source code, Jupyter notebooks, configuration files, requirements.txt

Documentation: Comprehensive README.md with setup instructions, usage examples, and architecture overview

Best Practices: Clear commit messages, .gitignore for large files, issue tracking



Technical Report (10-15 pages)

Structure: Abstract, Introduction, Related Work, Methodology, Results, Discussion, Conclusion, References

Key Elements: Clear problem formulation, detailed methodology, comprehensive results with tables/figures, critical analysis

Style: Academic writing, IEEE or similar format, properly cited references

Quality: Proofread, professionally formatted, publishable quality



Presentation Slides

Duration: 15-20 minutes (approximately 15-25 slides)

Content: Problem statement, methodology overview, key results, clinical implications, demo walkthrough



Demo Video (5-10 min)

Format: Screen recording with voiceover or talking head + screen share

Content: System walkthrough, key features

Design: Professional template, clear visualizations, minimal text, high-quality graphics

Delivery: Practice timing, prepare for Q&A, engage audience with demo

demonstration, real use case scenario, results visualization

Quality: Clear audio, smooth transitions, professional editing

Platform: YouTube (unlisted) or institutional platform, include captions for accessibility

Sample GitHub Repository Structure

```
medical-ai-project/
├── README.md
├── requirements.txt
├── setup.py
└── data/
    ├── raw/
    ├── processed/
    └── README.md
└── notebooks/
    ├── 01_data_exploration.ipynb
    ├── 02_preprocessing.ipynb
    └── 03_model_training.ipynb
└── src/
    ├── data/
    │   ├── preprocessing.py
    │   └── augmentation.py
    ├── models/
    │   ├── architecture.py
    │   └── train.py
    ├── evaluation/
    │   └── metrics.py
    └── utils/
        └── helpers.py
```

```
└─ models/
    ├─ checkpoints/
    └─ final_model.pth
└─ results/
    ├─ figures/
    └─ metrics.json
└─ docs/
    ├─ technical_report.pdf
    ├─ presentation.pptx
    └─ demo_video_link.txt
└─ tests/
    └─ test_model.py
```

Pro Tips

For Code Repository:

- Use meaningful variable and function names
- Add inline comments for complex logic
- Include unit tests for critical functions
- Create a detailed README with setup instructions and dependencies

For Technical Report:

- Start with an outline before writing
- Use high-quality figures and tables with clear captions
- Cite all sources properly (aim for 15-30 references)
- Have peers review before final submission

For Presentation:

- Practice multiple times, time yourself
- Use the "rule of one": one main idea per slide

- Include a compelling opening and strong conclusion
- Prepare backup slides for detailed Q&A

For Demo Video:

- Write a script before recording
- Use high-quality microphone
- Show real data examples, not just synthetic ones
- Include brief explanation of what's happening on screen

Ready to Begin Your Medical AI Journey?

Remember: Great projects start with clear planning, thrive on strong teamwork, progress through structured milestones, and shine with professional deliverables. Focus on creating meaningful clinical impact while demonstrating technical excellence. Your work has the potential to improve healthcare outcomes—make it count!

Questions? Reach out to your instructors early and often. Good luck! 