

Gantt Chart - Anti-Phishing System Project

Student Version (Solo Developer)

Project: Fraud Detection, Prevention and Reporting System for Malicious Websites

Developer: Solo Student

Duration: 14 weeks (~3.5 months)

Date: November 22, 2025

Visual Gantt Chart

LEGEND:

 Completed

 In Progress

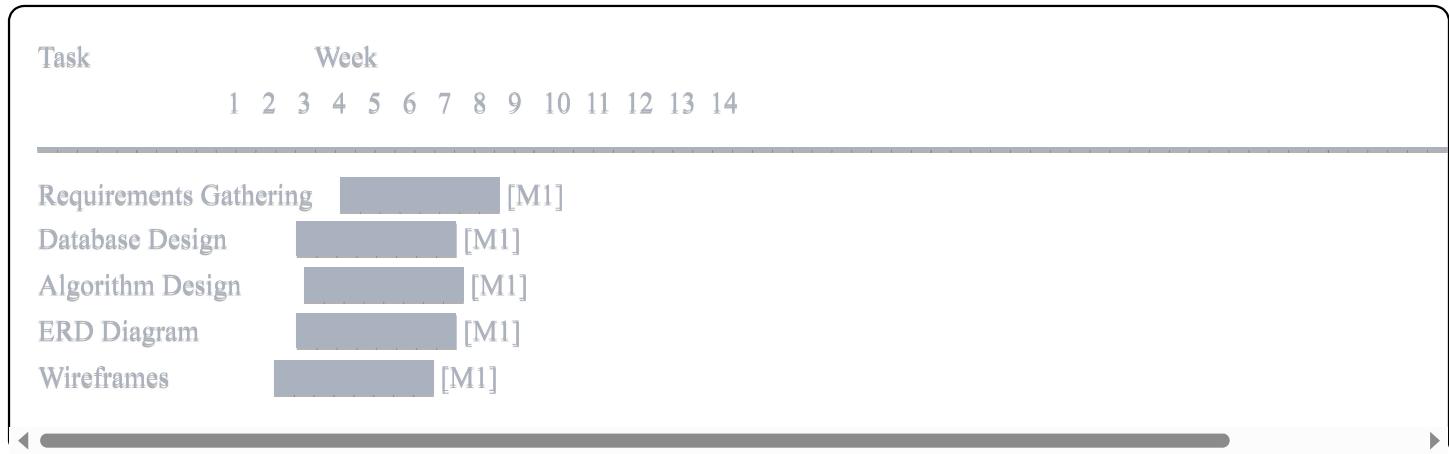
 Planned

[M] Milestone

Week: 1 2 3 4 5 6 7 8 9 10 11 12 13 14



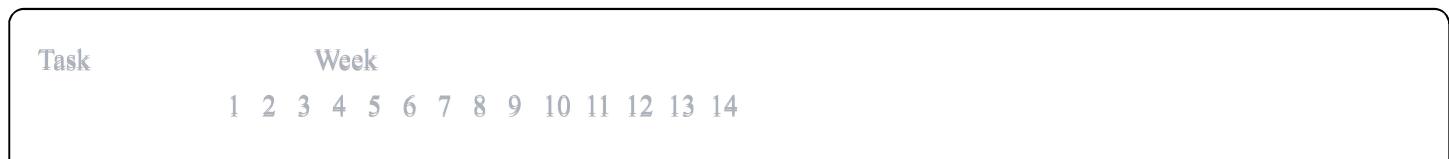
PHASE 1: DESIGN & PLANNING COMPLETED

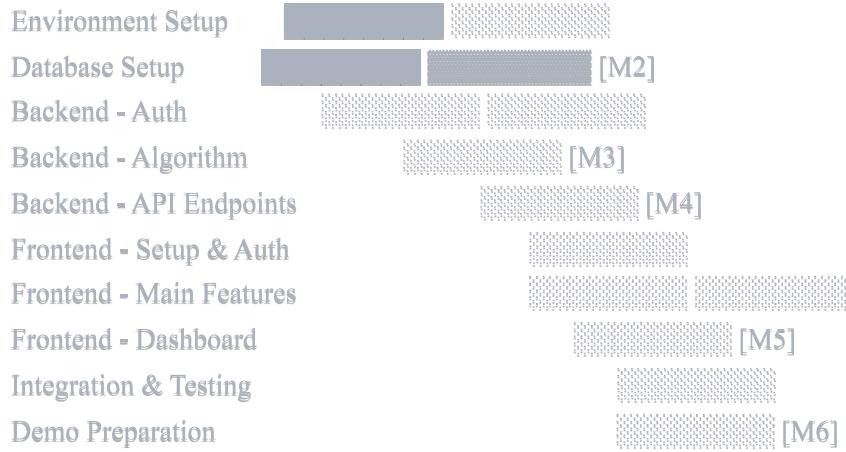


Status:  COMPLETED (Weeks 1-2)

Milestone M1: All design documents complete

PHASE 2: DEVELOPMENT





Week-by-Week Breakdown:

Week 3: Environment Setup

Week 3: [M2] [Progress Bar]

Tasks:

- Install Node.js, PostgreSQL, VS Code, Git
- Create Git repository
- Set up project folders

Week 4: Database Setup + Backend Foundation

Week 4: [M2] [Progress Bar]

Tasks:

- Create PostgreSQL database
- Execute database_schema.sql
- Import sample data
- Set up Express.js server

[M2] Milestone 2: Database fully set up with sample data

Week 5: Backend - Authentication

Week 5: [M3] [Progress Bar]

Tasks:

- User registration endpoint
- User login endpoint with JWT
- Role-based authentication middleware

Week 6: Backend - Core Algorithm ★

Week 6:

Tasks:

- Implement Levenshtein similarity
- Implement Jaro-Winkler similarity
- URL parsing and normalization
- Scoring system

[M3] Milestone 3: URL detection algorithm working

Week 7: Backend - API Endpoints

Week 7:

Tasks:

- POST /api/check-url (main feature)
- GET /api/check-history
- POST /api/reports
- GET /api/reports
- Test with Postman

[M4] Milestone 4: All core API endpoints functional

Week 8: Frontend - Setup & Authentication

Week 8:

Tasks:

- Create React App setup
- Login page
- Register page
- Basic layout (header, footer)

Week 9: Frontend - Main Features

Week 9:

Tasks:

- URL check page with input form
- Results display (score, classification, breakdown)
- Scan history table

Week 10: Frontend - Dashboard & Reports

Week 10: 

Tasks:

- Report submission page
- Dashboard with statistics
- Moderator page for report review

[M5] Milestone 5: Complete frontend functional

Week 11: Integration & Testing

Week 11: 

Tasks:

- Connect frontend to backend
- Test all features end-to-end
- Algorithm accuracy testing
- Fix critical bugs

Week 12: Demo Preparation

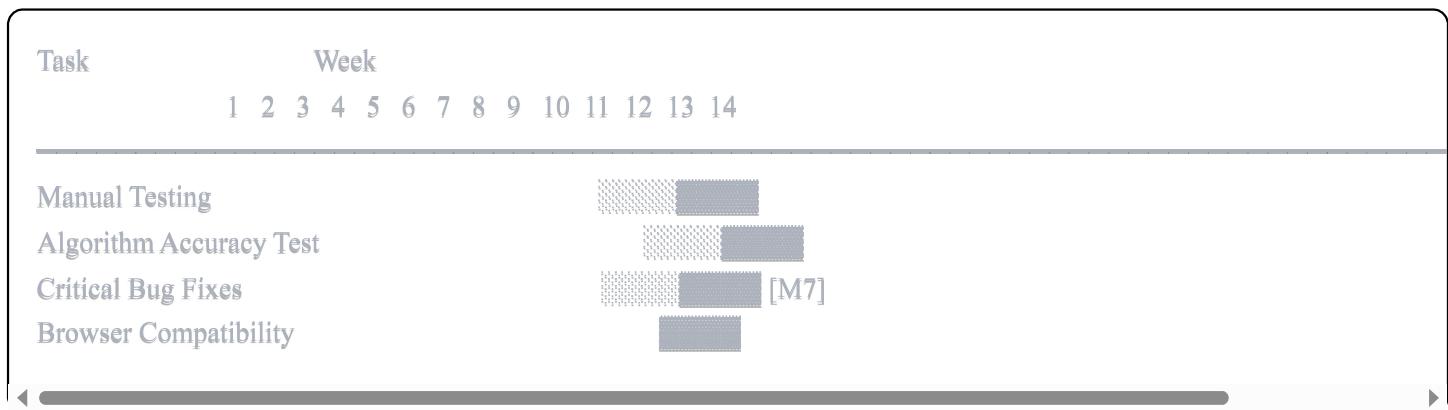
Week 12: 

Tasks:

- Prepare demo data
- Record demo video
- Take screenshots
- Final bug fixes

[M6] Milestone 6: Working prototype complete, ready for documentation

PHASE 3: TESTING (Simplified)



Week 11-12: Testing

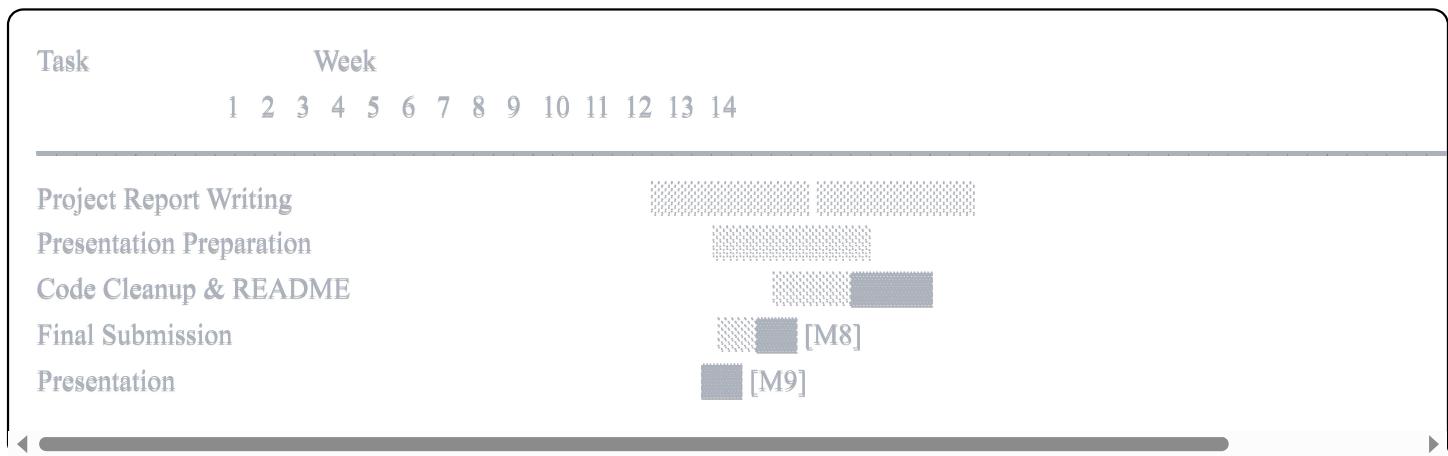
Week 11-12: [Progress Bar]

Tasks:

- Manual testing of all features
- Test with 30+ known phishing URLs
- Test with 30+ legitimate URLs
- Calculate accuracy percentage
- Fix critical bugs only

[M7] Milestone 7: Testing complete, system stable

PHASE 4: DOCUMENTATION & SUBMISSION



Week 13: Report Writing (Part 1)

Week 13: [Progress Bar]

Tasks:

- Write chapters 1-3 (Introduction, Literature Review, Design)
- Include ERD, algorithm diagrams
- Add references

Week 14: Report Writing (Part 2) & Presentation

Week 14: [Progress Bar]

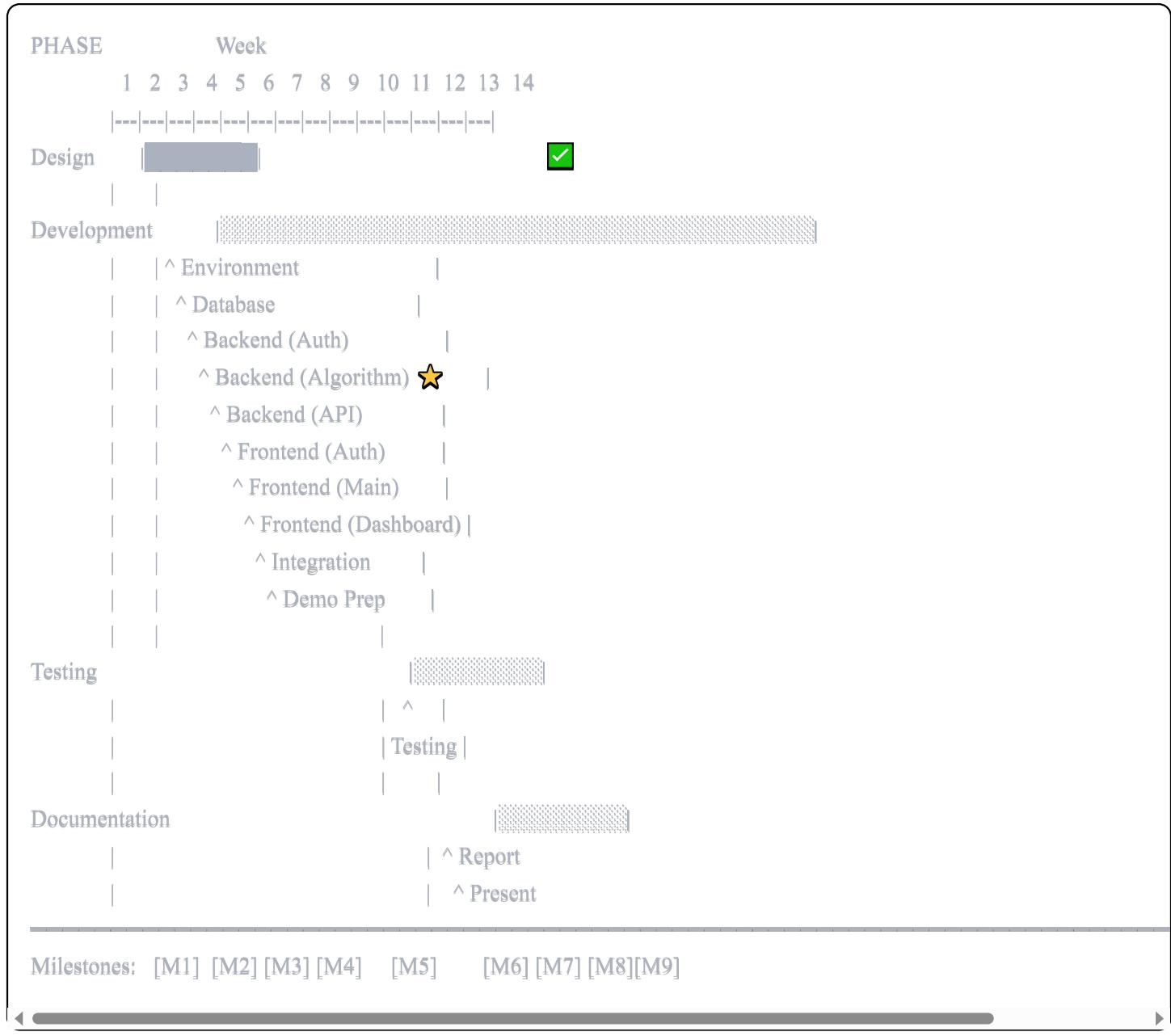
Tasks:

- Write chapters 4-6 (Implementation, Testing, Conclusion)
- Proofread and format
- Create PowerPoint slides
- Practice presentation
- Submit all materials

[M8] Milestone 8: All documentation complete, submitted

[M9] Milestone 9: Final presentation delivered

COMPLETE TIMELINE - BAR CHART VIEW



MILESTONE SUMMARY

Milestone	Week	Description	Deliverable
M1	Week 1-2	Design Complete	✓ Database schema, Algorithm doc, ERD
M2	Week 4	Database Ready	PostgreSQL with all tables, sample data
M3	Week 6	Algorithm Working	Core detection algorithm functional
M4	Week 7	Backend Complete	All API endpoints working
M5	Week 10	Frontend Complete	Full web application functional
M6	Week 12	Prototype Ready	Working system, demo prepared

Milestone	Week	Description	Deliverable
M7	Week 12	Testing Complete	Bugs fixed, accuracy measured
M8	Week 14	Documentation Done	Report + presentation submitted
M9	Week 14	Project Complete	Final presentation delivered <input checked="" type="checkbox"/>

CRITICAL PATH

The critical path (tasks that cannot be delayed without delaying project):

Design (M1) → Database Setup (M2) → Algorithm (M3) → API (M4) → Frontend (M5) → Integration (M6) → Testing (M7) → Documentation (M8) → Presentation (M9)

Total Critical Path Duration: 14 weeks

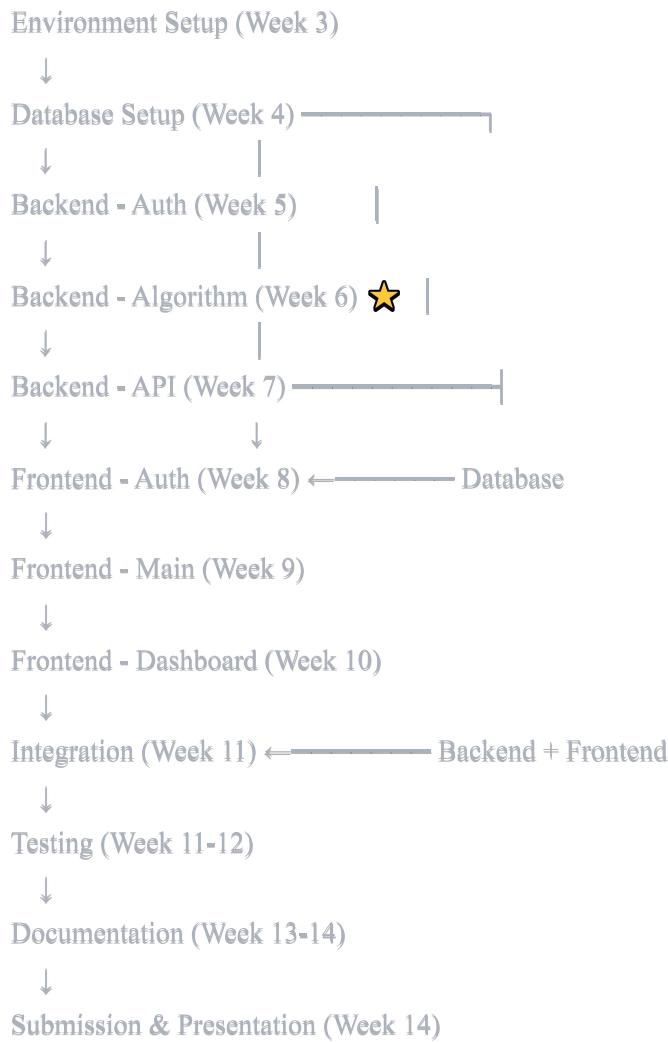
No slack time - stay on schedule!

RESOURCE ALLOCATION PER WEEK

Week	Hours/Week	Focus Area	Intensity
1-2	0 hrs	Already complete	<input checked="" type="checkbox"/>
3	10-15 hrs	Environment setup	Low
4	15-20 hrs	Database + backend start	Medium
5	20-25 hrs	Backend authentication	Medium-High
6	25-30 hrs	Algorithm (most important!)	High 
7	20-25 hrs	API endpoints	Medium-High
8	15-20 hrs	Frontend setup	Medium
9	20-25 hrs	Frontend main features	Medium-High
10	20-25 hrs	Frontend dashboard	Medium-High
11	25-30 hrs	Integration & testing	High
12	20-25 hrs	Testing & demo prep	Medium-High
13	30-35 hrs	Report writing (intense!)	Very High
14	35-40 hrs	Final report & presentation	Very High

Total: ~300-350 hours over 14 weeks
 Average: ~21-25 hours per week

DEPENDENCIES MAP



Key Dependencies:

- Frontend CANNOT start until backend API is working (Week 7 done)
- Integration CANNOT start until both frontend and backend are done
- Documentation should reference working demo

RISK BUFFER

Planned Duration: 14 weeks

Recommended Buffer: +1-2 weeks for unexpected delays

Safe Deadline: Week 16 (gives you 2 weeks buffer)

Built-in buffers:

- Week 11-12: Testing phase can absorb small delays

- Week 13-14: Can compress documentation if needed
 - Browser extension: Already optional (can skip)
 - AWS deployment: Already optional (can demo locally)
-

WEEKLY CHECKLIST

Week 1-2: COMPLETED

- Database design
- Algorithm design
- ERD diagram

Week 3: Environment Setup

- Install Node.js, npm, PostgreSQL, VS Code, Git
- Create GitHub repository
- Set up project folder structure
- Initialize backend (Express.js)
- Initialize frontend (Create React App)

Week 4: Database + Backend Start

- Create local PostgreSQL database
- Execute database_schema.sql
- Import sample phishing URLs (50-100)
- Import legitimate brand list (20-30)
- Set up Express.js server
- Configure database connection

Week 5: Backend Authentication

- Implement user registration
- Implement user login with JWT
- Create authentication middleware
- Test with Postman

Week 6: Backend Algorithm CRITICAL

- Implement Levenshtein similarity function
- Implement Jaro-Winkler similarity function
- Implement URL parsing
- Implement URL normalization
- Implement domain analysis

- Implement subdomain analysis
- Implement scoring system
- Test algorithm accuracy

Week 7: Backend API

- POST /api/check-url endpoint
- GET /api/check-history endpoint
- POST /api/reports endpoint
- GET /api/reports endpoint
- PUT /api/reports/:id/status endpoint
- GET /api/dashboard/stats endpoint
- Test all endpoints with Postman

Week 8: Frontend Auth

- Create React app structure
- Create login page
- Create register page
- Create header/footer layout
- Connect to backend auth API

Week 9: Frontend Main Features

- Create URL check page
- Create results display component
- Create scan history component
- Test URL checking flow

Week 10: Frontend Dashboard

- Create report submission form
- Create dashboard with statistics
- Create moderator review page
- Add basic styling (Bootstrap)

Week 11: Integration & Testing

- Connect all frontend pages to backend
- Test complete user flows
- Test algorithm with real URLs
- Calculate accuracy percentage
- Fix critical bugs

Week 12: Demo Preparation

- Create demo user accounts

- Prepare demo data
- Record demo video (5-10 min)
- Take screenshots
- Test presentation flow

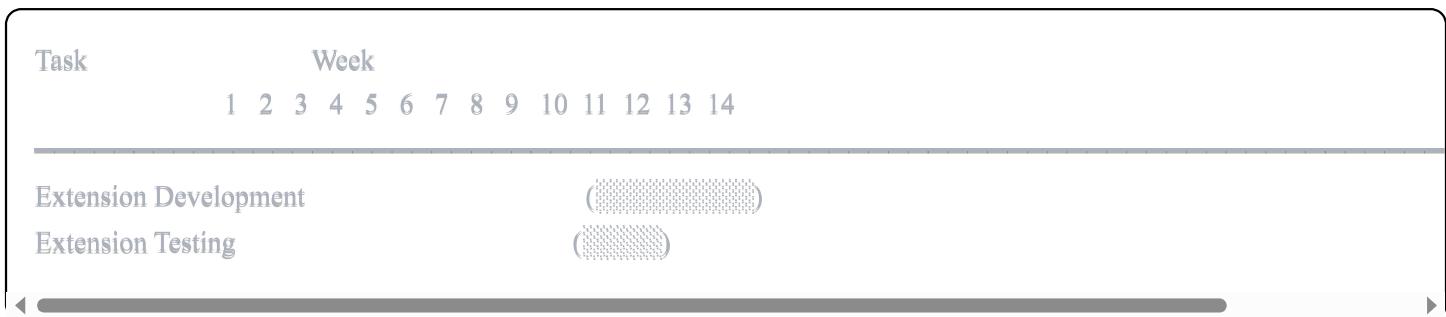
Week 13: Report Writing (Part 1)

- Write Chapter 1: Introduction
- Write Chapter 2: Literature Review
- Write Chapter 3: System Design
- Add diagrams (ERD, architecture)

Week 14: Report Writing (Part 2) & Presentation

- Write Chapter 4: Implementation
 - Write Chapter 5: Testing & Results
 - Write Chapter 6: Conclusion
 - Format and proofread report
 - Create PowerPoint slides (15-20)
 - Practice presentation
 - Submit all materials
 - Deliver presentation
-

OPTIONAL: BROWSER EXTENSION (If Time Permits)



Only attempt if:

- Backend and frontend are done early (by Week 10)
- You have 1-2 extra weeks
- All core features are working well

If skipped: Mention in report as "planned future feature"

STUDENT TIPS FOR STAYING ON SCHEDULE

Time Management:

1. **Set daily goals:** "Today I will complete user registration endpoint"
2. **Work in 2-hour blocks:** Pomodoro technique (25 min work, 5 min break)
3. **Track progress:** Check off tasks daily
4. **Weekend catch-up:** Use weekends if you fall behind during week

Avoid Scope Creep:

1. **Resist adding features:** Stick to the plan
2. **"Future work" list:** Write down ideas but don't implement now
3. **Focus on MVP:** Minimum Viable Product first, polish later

When Behind Schedule:

1. **Week 6-7:** Skip browser extension
2. **Week 8-10:** Use Bootstrap templates, simplify UI
3. **Week 11-12:** Reduce database to 12 core tables
4. **Week 13-14:** Focus on report content, not perfect formatting

When Ahead of Schedule:

1. **Polish UI:** Make it look professional
 2. **Add charts:** Use Chart.js for dashboard
 3. **Try browser extension:** Add extra feature
 4. **Deploy to AWS:** Show cloud skills
-

SUCCESS METRICS BY WEEK

Week 6 (Algorithm):

- Can detect at least 70% of test phishing URLs
- False positive rate < 10%
- Average processing time < 200ms

Week 7 (Backend):

- All API endpoints return correct responses
- No critical errors in Postman tests

- Database queries execute < 50ms

Week 10 (Frontend):

- Can complete full user flow without errors
- UI is responsive on desktop and mobile
- Results display is clear and understandable

Week 12 (Integration):

- Working demo from start to finish
- No crashes during typical usage
- Demo video looks professional

Week 14 (Documentation):

- Report is 40-60 pages with proper formatting
 - All diagrams included and referenced
 - Presentation flows smoothly in 15-20 minutes
-

TOOLS FOR PROJECT MANAGEMENT

Recommended (Free):

1. **Trello:** Create board with columns for To Do, In Progress, Done
2. **Google Sheets:** Track weekly hours and tasks
3. **GitHub Projects:** Built-in project board
4. **Notion:** All-in-one workspace

Simple Approach:

- **Checklist in this document:** Just check boxes as you complete
 - **Calendar:** Mark milestones on Google Calendar
 - **Daily log:** Keep simple text file of what you did each day
-

FINAL TIMELINE SUMMARY

PROJECT TIMELINE (14 WEEKS)

- Week 1-2: Design Complete (Already Done)
- Week 3-7: Backend Development (5 weeks)
 - Environment, Database, Auth, Algorithm, API
- Week 8-10: Frontend Development (3 weeks)
 - React setup, URL check page, Dashboard
- Week 11-12: Testing & Demo Prep (2 weeks)
 - Integration, Bug fixes, Screenshots
- Week 13-14: Documentation (2 weeks)
 - Report writing, Presentation, Submission

TOTAL: 14 weeks = ~3.5 months = 1 semester

Remember: This is a **guideline**, not a strict rule. Adjust as needed, but try to keep major milestones on track!

Good luck with your project!

Document Version: 1.0

Last Updated: November 22, 2025

Status: Active Project Plan

Next Action: Start Week 3 - Environment Setup!