FREE CHECKLIST

FYP / Capstone Scope Checklist for Singapore Students

The shortcuts I tell every team in week one, and the week-by-week milestones that separate the FYPs that ship from the ones that collapse in week 22.

THE FIVE RULES OF SCOPING

1. **Pick boring tech.** Postgres, React, Python, Express, Spring Boot. The marker doesn't reward novelty, just delivery.
2. **Cap features at five for the first demo.** Cut everything else into a "nice to have" list you may never get to.
3. **Match the stack to your team's existing skills.** Don't learn Rust during an FYP. Learn architecture instead.
4. **Write the problem statement in one paragraph.** If you can't, the FYP isn't ready to start.
5. **Decide a weekly meeting cadence in week one.** 30 minutes, same time, every week. Skip if there's nothing, never skip "because we're busy".

STACK CHOOSER

Web app

- Frontend, React + Vite or Next.js
- Backend, Node + Express, Python + FastAPI, or Java + Spring Boot
- Database, PostgreSQL (almost always)
- Auth, NextAuth or Auth0 free tier
- Hosting, Vercel + Render + Supabase, all free

Mobile app

- React Native or Flutter (whichever the team knows)
- Backend same as web
- Skip native iOS / Android unless the FYP is about platform features

Data project

- Python + pandas + scikit-learn
- PyTorch only if deep learning is the FYP point
- Jupyter for exploration, scripts for the final pipeline

WHAT TO NOT DECIDE IN WEEK ONE

- Microservices vs monolith. Monolith. Always.
- Whether to use TypeScript. Yes. Just yes.
- The CSS framework. Tailwind, or whatever your frontend folks already know.
- The CI/CD pipeline. Just deploy on push. Anything fancier is procrastination disguised as engineering.

- Authentication strategy. Use the framework's default and move on.

WEEK-BY-WEEK MILESTONES, 24 WEEK PROJECT

By week 1

- Problem statement in one paragraph, written down.
- Five core features picked, written down.
- Stack chosen, written down.
- Repo created, main branch protected, weekly meeting time set.

By week 6

- One core feature working end-to-end, demoable to a stranger.
- 50+ commits across the team.
- Deployed staging environment that anyone can access.
- Clear written list of features still missing.

By week 12 (mid-point demo)

- 3 of the 5 core features working.
- Integration tests, even basic ones.
- Mid-point report rough draft started.
- Outstanding risks called out openly with the supervisor.

By week 18

- All 5 core features working.
- Stop adding features. Polish only.
- User testing or at least a friend-test.

By week 22, last 2 weeks

- Bug fixes, no new features.
- Documentation, dev guide, user guide.
- Demo script written and rehearsed.
- Slides for final defence drafted.

DEMO DAY CHECKLIST

- Dry-run the demo at least twice the day before.
- Have a fallback recorded video, in case the live demo hangs.
- Print or screenshot key code sections you might be asked to defend.
- Bring a pre-loaded laptop, not "I'll just run it on the projector PC".

- One sentence ready for "what would you do differently".

Need a second opinion before you commit to a stack?

Send the brief, I'll give you a stack recommendation in 10 minutes and save you weeks of regret.

codingsolutions.dev/services/fyp · Telegram, @IsaacNgCS