

Final project (FP)

Duration: 7 weeks

The final project demonstrates the learners ability to apply and assimilate the various skills acquired throughout the course. Furthermore the student will learn to structure larger applications and experience all phases of a software project. The main focus of this module is to work in a team and collaboratively create and improve a software application.

During the final project phase, students work in teams on a real-world project. During the weeks of the project, the groups go through all the phases, from the ideation of a concept to its implementation and presentation.

Week Distribution

Week	Topic	Content
Week 1	"Ideation week"	Stack and MVP definition, individual pitch
Week 2	"Design"	Mockup and User Stories
Week 3	"Development"	Bootstrap project
Week 4	"Implementation"	Core features development
Week 5	"Review"	Feature freeze
Week 6	"Finalize"	Documentation and Deployment
Week 7	"Present"	Prepare presentation

Key points

- Pitching will take place at the end of the 1st week
- Attendance on the day of pitching is compulsory and whoever will fail to be present will miss the chance to work on the final project.
- After the pitching, we'll have a vote to build the teams but the teachers will finalize the groups.
- Each group will have a minimum of 3 and a maximum of 5 participants.
- Solo-project will not be allowed to present in the final-event.

Teamwork

The team will share responsibilities but will define roles

- Project Manager
 - overall responsibility for the successful implementation of Agile methodology (Standups Meeting-Retrospectives-etc)
- Repository Maintainer
 - responsible for enforcing process in the GitHub project repository
- Lead FE
 - Frontend web development
- Lead BE
 - Backend web development

Tools

- Github Repository: source code version control
- Github Project: Dev and issues / Project Management
- Figma: UI prototyping
- Trello Board: Project Management
- ■ *

1 – Ideation week

Week plan

- project phase intro
 - outline timeline for the final project
 - outline requirements like team size, team member contribution, usage of version controls system
 - final project presentation
 - give important notes on what to pay attention to during the entire process
- find project ideas (Mon, Tue)
- prototype ideas either on paper or coding (Tue, Wed)
- form teams (might be good to make sure each team has some skills (FE, BE))
- learn a bit about Agile (Tue, Wed)
- create feature list and project roadmap (what we want to achieve by the end of the 8 weeks if everything goes as planned) (Wed, Thur)
- create issues / tickets and prioritize (for each project) (Wed, Thur)
- first sprint planning: pick items for 1st sprint (Wed, Thur)

At the end of week 1 (ideation week) each team should have:

- a project 1 liner: explain the project in 1 sentence, so anyone who is not familiar with the project can understand what it is all about.
 - Service to provide Hotel clients with web presentation and dashboard management.
 - Plutone is an experiment that aims to bring audio and visuals together, creating an interactive soundscape experience through the browser.

- set up project board: we are using GitHub projects (even though it offers only Kanban, so we have to simulate Scrum). Items on the board are linked to issues in the repo.

The screenshot shows a GitHub project board for 'Soil Data V1'. The board is organized into three columns: 'Backlog' (13 items), 'In Progress' (6 items), and 'Ready to deploy' (1 item). Each column has a header with a pencil icon for editing and a plus sign for adding new items. The 'Ready to deploy' column contains a single item: '[Data] Soil data collection scripts' (opened by Vijay). The 'In Progress' column contains several items, including 'Crawl tractor engine data (John Deere)' (opened by Melinda) and 'Performance updates for data script' (opened by Sam). The 'Backlog' column contains items such as 'Collect satellite data and deliver to farmers' (added by Sam), 'Launch Plan' (added by Sam), 'UI for accessing app on tractor screen' (added by Vijay), 'Fix CSV rendering' (added by Emily), 'Share farms' soil moisture data' (added by Eddie), and 'File sharing permissions' (added by Fabian). A tooltip for the 'Share farms' soil moisture data' item indicates it is linked to a 'New doc editor (@johndoe)'. A dashed box highlights the 'Ready to deploy' column.

- defined long term goals / features, broken down into steps (tickets)
- setup branching structure (optional)
- setup deployment pipeline (optional)
- as a teacher, collected all links to each repo and project board

Example project board:

<https://github.com/users/MScully88/projects/1#card-23584551>

Inspiration

- View few inspiring projects and discuss them together
 - From previous Final Project
 - From the internet
 - Inspiration

Brainstorming (the whole class or within a team)

- Generate ideas, without judgement: it is important to separate idea generation from idea feasibility
- Avoid having an idea owner, that pitches an idea and people join around them and their idea, ideas should be malleable
- You could work the whole class together
 - new groups may form around ideas, instead of groups based on friendship
 - ideas are not restricted to the people in your team
 - can combine ideas

How to think of ideas:

- turn a feature into a standalone product
- solve a problem from your every day life
- find a user group
- empathize first and foremost
- pick a technology you want to learn
- do something fun
- create a clone of product you like
- data insights

Process:

- generate ideas, without judgement
- write on post its and post them on the wall
- vote / classify by what seems interesting

Idea feasibility

After the brainstorm, some ideas will be discarded almost instantly, others need to be challenged. At the end of idea feasibility session, teams should have selected one idea.

Process:

- it is okay to have top 3 ideas after the brainstorm session

- if no ideas survive, you will need to go back to brainstorming
- challenge idea(s) -
 - it is important to ask clarifying questions, instead of judging ideas as not feasible
 - discourage people from not giving ideas any chance
 - clarify who are the users and what are their needs (in order to add features or to get rid of features)
 - ask to elaborate features and how things would work, but without going into too much details. It should be pointers into what directions to research further that can be assigned to the students to pursue.
 - clarify what problem it solves (does it solve any or too many)
 - how do you think to solve the problem

The goal is not to discard ideas, but to point towards the tasks that need to be done, and to narrow down scope, make the product more focused and clean and give students the construction how to challenge their ideas on their own.

Based on the answers of the clarifying questions create tasks for the students: research API, think of techniques to solve the problem, do a prototype, on paper or coded.

It is important to point out, that when building a product, you don't have to cover every aspect.

You can build a proof of concept, for example mocking all the usual features (like logging in) but focusing on one.

Idea selection

It can be difficult to settle down on an idea.

It is good to define the outcome of the project for each the team:

- learn something new
- have a cool product
- have fun
- get a job

For example, it might be important to apply for jobs, then maybe project should be a bit more relaxed with fixed schedule.

2 – Design week

Week plan

- Visualize your Concept
 - Envision, test and validate project idea with quick wireframes and detailed mockups.
 - Create user stories to fit our user personas into the context of the product design
 - Define requirements and functionalities
-

3 – Development week

Week plan

- Bootstrap project
 - Setup your development environment
 - Setup your GitHub project and management tools
 - Create tasks backlog
 - Start coding basic features
-

4 – Implementation week

Week plan

- Core features development
 - Implement core MVP features
 - Application testing
 - Create documentation
-

5 – Review week

Week plan

- Complete MVP features
 - Implement additional features
 - Application testing
 - Integrate documentation
 - Deploy project
 - Feature freeze by the end of the week
-

6 – Finalize week

Week plan

- Prepare for launching
 - Create presentation slides
 - Integrate documentation
 - Deploy project
 - Test deployment
-

7 – Present week

Week plan

- Prepare presentation
 - Complete Slides
 - Publish Documentation
 - Rehearsal final event
 - Presentation
-



During work weeks

Mon, Tue, Wed

Students should work on tasks. Make sure to check in with them and guide them into structuring work and getting them unstuck.

Ask if they have tasks, if they know what tasks they need to be working on and in what order.

Thu

- Review

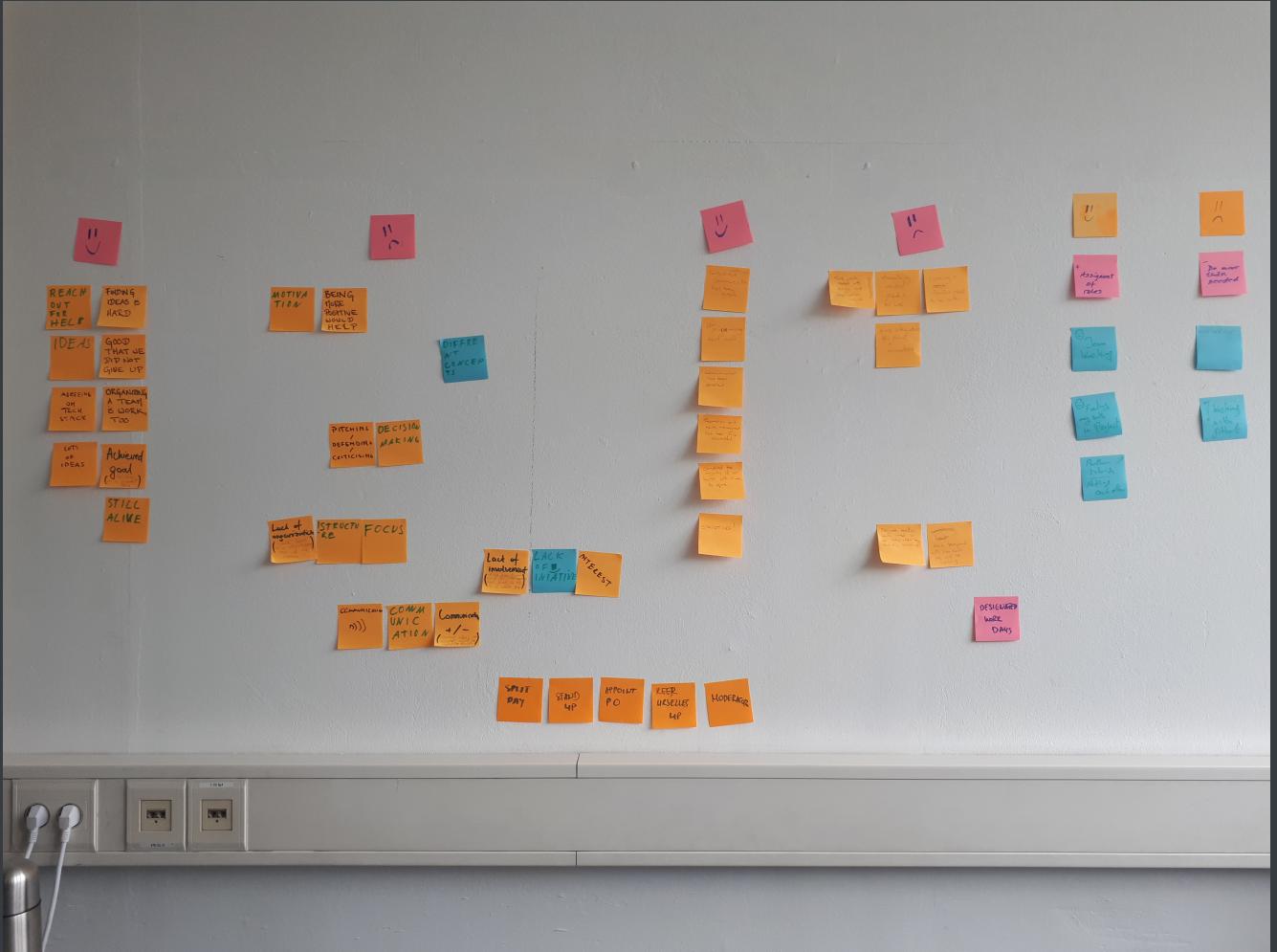
Each team shows a demo of what was achieved in the previous week ideally in sync together with the project board and deployed.

Project management, board setup is also work and can be presented.

Ask clarifying questions about tech used, problems encountered.

The idea of the demo is to share knowledge among the students and make sure teams are progressing.

- Retrospective (optional)



The whole class together spends 10 min to write down what went well, what didn't go well in their team.

It is good to have at least 1 positive item.

Use markers and use post-its. Write one item per post-it.

This is important to keep it short and readable.

For each team, each team member adds their notes to the wall and explains briefly what they had in mind.

No judgement, but clarifying questions are encouraged.

Once all teams are done, decide with each team what issues they want to tackle in the next sprint. Do not choose more than 3.

Next, generate ideas how to fix these issues - the solutions must come from the students and must be actionable.

For example:

low motivation, feeling down -> how can we support each other? -> we need an action -> allow everyone to say - "hey don't be a downer" or simply don't engage in pity talks.

lack of focus -> appoint a different team member for moderator in the team for each meeting.

- Refinement

Plan tasks for next week, refine bigger tasks, possibly assign tasks to team members.

Teacher's role during the Final Project

Topic	Content
"Ideation week"	Manage potential issues (e.g. "technology stack")
"Project Workflow"	Advise and supervise the project structure before kick-off. Implement and use a proper workflow
"Design"	Review UI/UX. Appealing User Interface and smooth User Experience
"Project Management"	Attend the team meetings, supervise the process, 1:1 with PMs
"Planning-Workflow"	Bug tracking, good practices
"Presentation"	Train the teams for the presentation day
"Present"	Prepare presentation

- Teachers will not interfere in the development: Teachers will not dictate code line-by-line, provide code snippets or implement features.

How to take help from the teachers

In case of working on the final projects from home effective communication is extremely important and teachers need to be involved in your daily activities in the following ways:

- Share your github repository.
 - Create a zoom/hangouts meeting room and share the link. - Use the same link everyday.
 - Create a slack channel/workspace in order to share documents, resources and for important announcements.
 - If you need assistance from the teachers, send them a message on slack and they'll join your meeting.
-

General notes for students

Make sure students are aware of this list and check in with them from time to time.

- Try and document everything regarding the process

This will help you with creating the final presentation, providing you with lots of material.

An example of project documentation: <http://ericasalvetti.com/portfolio/liefery/>

List down:

- what are your challenges
- why did you make a certain choice
- what were the other options, why did you discard them
- take photos of hand drawn sketches
- make videos / screen recordings of early stage prototypes
- record video demo
- take selfies if you like
- Make sure you always have a working state of your project
- Keep everything in version control system

- Prototype
 - on paper
 - small coding prototypes (try different approaches, different tech to check feasibility)
- Prototyping is cheap, it can save you lots of time and effort going in the wrong direction.
- Give the project a name, possibly a logo (but not required)
 - Possibly create a product page to advertise the project
 - If it is an app create page that allows for download (cause it is hard to get it to the app / play store).
 - Outline unique selling points, present the team etc.
 - If you feel like it, go to talk to some real users (user research) and write down your findings
 - This can be very helpful determining product features, and testing your ideas.
 - Make sure you define who your users are
 - You can go as far as doing personas, or you can think of a more general group like businesses and try to think of what their needs might be.
 - This way of thinking will help you define what features to develop and what problems to solve.
 - Record a demo in case a live demo is not possible / is not working etc.
 - Use autofill for forms, or automate the boring parts of your demo.
 - Make sure you have project page documentation (to base later the project roadmap)
 - This is not technical documentation, it is about the product idea.
 - Try not to change things too much after you plan the project so you don't end up constantly changing stuff (not too Agile but the idea is that people should commit and stick to their commitment)
 - Leave time at the end for bug fixing and presentation creation and testing
 - Consider writing a blog post how you build your final project (you can post it on FreeCodeCamp, Medium, or on the product site)

Application process

During the final project phase students have to continue to work on the application process:

- Finalizing their portfolio, GH repositories, CV
- Apply for jobs and internships
- Complete code-challenge
- Preparing for technical interviews (prepare to present one of their projects)

Presentation day

- Check if all tech needed is in place (sound system that you can control for example for music exploration project, or available ports)
- Check with organizers for program / order of events planned for the day and communicate it to students
- On presentation day go earlier to space with students so they can explore the place and get accustomed
- Opt for clip on microphones instead of handheld (as it will be hard to live code or click around while holding a mic)
- Think to dress up as a team (wear same tshirt with logos, or bandanas)
- Use sticky labels to use name tags
- Dress up to indicate same team you could use bandanas, or same color t-shirts, you could print the logo on your project on t-shirts as well, or you could wear fancy special occasion clothes, anything that will make this day memorable for you

Guide To Final Presentations

Document to share with students

Prepare a presentation following these recommended steps:

1. Introduce your project
 - a. What is the idea of your project or the motivation behind it (the audience loves story telling)?
 - b. Which problems does it solve?
2. Present your team

- a. Show your names
 - b. If you have - add photos to the presentation (beneficial for the companies to identify you afterwards for networking)
 - c. Did you split tasks? Who had which role in your team?
 - d. How did you cooperate within your team?
3. Which Technologies have you used? Why did you use these specific technologies?
 4. How did you implement them?
 - a. Show how the website runs (focus on main functions, rehearse this carefully in advance)
 - b. What features does it have?
 - c. What makes it stand out?
 5. Talk about challenges.
 - a. What kind of challenges did you face during the project phase? (Don't mention points that don't prove you as a good team-player, focus on technical challenges)
 - b. Is there anything you would improve next time?
 - c. What have you learned during the process?
 6. Elevator pitch
 - a. Self-marking and speak what kind of problem you can solve.
 - b. Talk about your skills.
7. Participation

Participating in both trial and real presentations are mandatory even if some of the projects are not completely ready.