Group Project - HappyData

Overview

You are building **HappyData**, a web-based data visualization platform that helps users explore how different **development indicators** relate to **human well-being**. Your project will combine data from three key real-world sources:

1. **World Happiness Report**: https://data.worldhappiness.report/map

2. **World Bank Indicators API**:
   https://datahelpdesk.worldbank.org/knowledgebase/articles/889392

3. **World Bank Country & Lending Group Classification**:
   https://datahelpdesk.worldbank.org/knowledgebase/articles/906519

By building this platform, you will:

- Learn to access and work with global datasets.
- Apply software engineering principles in requirements, design, and implementation.
- Understand how development, equity, and sustainability can be explored through data.
- Reflect on the ethical and interpretive limitations of comparing countries.

Target Users of HappyData

- College students and researchers exploring development and sustainability.
- School students learning about sustainability and well-being
- Curious citizens interested in global happiness and fairness.

Product capabilities

Following visualizations should be available:

1. **Country Indicator Trends**
   Given a country and an indicator, show how the indicator has changed over time.
2. **Indicator vs. Happiness Comparison**
   Given a country and an indicator, display both trends together to visualize their relationship.

3. **Regional Happiness Distribution**
    Visualize happiness levels across World Bank regions.
4. **Regional Indicator Snapshot**
    For a selected year and region, compare values of a chosen indicator across countries.
5. **(extra-credit) Dashboard page of relevant indicator vs. happiness comparison for India**
    Identify a few indicators that correlate (positively or negatively) with happiness for India, and offer them as standard dashboard visualizations which can be filtered by years. For example, if poverty is one of your chosen indicators, have a dashboard widget for poverty vs. happiness over the years. Similarly for all chosen ones. Filtering by year range should change all the India indicators to show the selected range.

What you will build

Your team will play two types of roles. In Phase 1, your team will play the role of a product team that gathers the requirements and builds a specification for the product to be built. In Phase 2, your team will play the role of an engineering team that builds a product based on the specification provided. Once the product is built, you will also do an ethical reflection of the product you built.

Note that this product can take up a lot of time if you do not carefully scope your work. So spend time defining the scope that you can tackle and focus on completing that. You can control the scope, either when you are writing specifications (and prioritizing use cases), or when you commit to build the solution (in the design document). Clearly call out the scope of the product.

Phase 1

Your team plays the role of a product owner/product manager.

Goal in this phase is to apply SE principles for requirements gathering and modelling and produce a specification that can be built. Here is the typical tasks you will need to complete (not necessarily in this sequence, decide the specifics within your team):

1. Create one or more persona (and their details) that your product will target.
2. Create a few epics that will be delivered, and the list of user stories under these epics (talk to real people who fit the persona, also use AI to play the role of a given persona and get requirements)
3. Prioritize your user stories and produce the product backlog
4. Analyze the requirements (the user story list) and model the function, data and behavior of the proposed product.
5. **Create a specification document.** It should include problems and goals, target user persona, key user needs, the user story list to be worked on, proposed functional

solution, its architecture, and any assumptions or constraints for the solution.

Submissions

1. JIRA links to epics and the stories linked to these epics.
2. Specification document (2-3 pages)
3. A retrospective (similar to sprint retrospective) document that captures lessons learned (good, bad, ugly) during Phase 1. (1-2 pages)

Phase 1 needs to be completed before mid-term.

Phase 2

Your team plays the role of an engineering team that delivers the software based on the specification document.

Goal in this phase is to apply SE principles for design, implementation and testing, and produce a working product that conforms to the specification.

Here is the typical tasks you will need to complete (not necessarily in this sequence, decide the specifics within your team):

1. Estimate the work and define the scope you can deliver in the timeframe available.
2. Design the screens, choose APIs, model data and behavior to better understand what needs to be built as part of the solution proposed in the specification.
3. **Create a [design document](#)** that includes the details required to do the implementation and testing - screens, APIs, data schema, workflows and behavior,
4. Decide on the implementation strategies - identify components and modules to be built, select the libraries, tools, and technologies to use. Select the technology stack for the development.
5. Implement the features (a working prototype at least) and deploy it after ensuring quality and conformance to requirements.

Submissions

1. Design document (3-5 pages)
2. Product release note (what features and use cases are working) - 1 page
3. Demo video that highlights key scenarios and features as built
4. Link to the deployed and working web application
5. A retrospective (similar to sprint retrospective) document that captures lessons learned (good, bad, ugly) during these Phase 2 work. (1-2 pages)

Ethical Reflection

Submit a brief reflection about the product you build. Use these questions to guide your write-up (one way is to have the answer to each of these questions as a distinct paragraph, mix-and-match is also fine).

- What indicator(s) aligned most with happiness? Where did alignment break down?
- What are the limitations of comparing development across countries?
- What biases might exist in the data?
- How can data visualization itself influence perception?

Submission

1. Write-up on your ethical reflection about the product you built (1-2 pages)

Technology Choice

Choose what lets you build quickly, keep the capabilities of your team members in mind. There are multiple options available for you to choose from. You should do a quick prototyping exercise before selecting if you are selecting something new to you.

1. Use a simple framework like Vue or plain HTML/CSS/JS. Only build front-end that calls APIs, does not have any kind of persistence.
2. Use Firebase or similar app development platform and include persistence and authentication capabilities.
3. Build your own backend using node/python/java/C# and a database of your choice, and start from some boilerplate code to get you started.

If you are doubtful about your technology choice, set up some time to discuss with the instructor.

Project Viva

You will be required to schedule a 30 minute interaction with me where you will present to me your progress, your plan for the remainder of the work, and answer any questions I may have. This is to ensure I can understand your work better and also provide you with any guidance you need. This is in addition to any ad-hoc discussion that you need with me about any aspect of your project.

Timelines

1. Phase 0 (Initial submission): Your understanding of the problem and the scope you will work on (1-page) - Jun 7, 2025

2. Phase 1 submissions are due on Jul 7, 2025 .
3. Phase 2 submissions are due on Aug 30, 2025 .
4. Ethical reflection submission is due on Aug 30, 2025 .
5. Project viva should be scheduled for 30 min, between Jul 15, 2025 and Aug 29, 2025 .

Evaluation Criteria for Group Project

Here are the criteria that will be used to evaluate your project.

1. Completeness and clarity of artefacts (specs, design, user stories, write-up)
2. Team collaboration and submission quality
3. Correctness and usability of implemented features
4. Quality of end product - visualizations and insights
5. Ethical reflection and awareness of limitations
6. Original thoughts and ideas
7. Evidence of good use of AI

Tips for working on the project

You do not need to follow each of them, these are just ideas to help you operate more efficiently:
1. Use the agile development process and practices we have discussed in class (and what you can read up).
2. Identify tools that you will use in various stages of your work - persona development, component design, interaction design, development, testing, etc. Take the opportunity to learn one or more tools. Each of you may also want to focus disproportionately on one tool - one person, one tool, so that the team gets the benefit from multiple tools, and each of you learn a new skill.
3. Spend some time on technology selection - do some prototyping and try to use what you know already. If you do not have any developer in your team, or you are not confident of development, consider using no-code/low-code platforms which will do most of the work for you if you can give sufficiently clear instructions. Explore them and finalize your technology.
4. After identifying requirements, identify a slice of the work that can be completed and plan to build it as your first iteration and deliver a usable release.
5. Spend time writing the specifications, it will help you use the AI in a more effective way - your prompts will be clearer, and you will know when the output does not match what you need. Same goes for design.
6. Each of you should keep a journal or notes as you work. This is a learning assignment, and you will learn best when you can reflect on your work later. This will also help you retrospect together in your team and produce the retrospective document.

7. Each of you should identify one process activity that you will focus on. For example, one of you may focus on the requirements phase, another may focus on interaction design, still another may focus on technology and construction, etc. More than one person can focus in a given phase, as long as all phases are covered (so some of you may have to cover more than one, though not with the same focus). Prioritize your learning this way.
8. Plan to meet regularly as a team and sync-up so that you are not bottlenecked or waiting on others' work. Also use JIRA to track your progress and ensure visibility. Meeting can be done on whatsapp or even email if in-call meetings seem infeasible. Specifying some time window for meeting (and actually meeting) is more important than the mechanism.
9. Scope your work. Estimate how much time each of you have available to do the work (try to get as much time as you can, most of your learning from this course is predicated on this project) and distribute the tasks appropriately. Keep doing this exercise periodically - if you use Scrum, every sprint boundary you get to change responsibility and tasks.

AI Usage

Many of the tasks listed above can be done better with the help of AI. However, as mentioned before, use AI to brainstorm, discuss and learn, not to generate responses that you submit directly (or with only small changes). A reminder of the expected usage:
1. Have a dialogue with AI when you need to come up with an idea or solution. For example, document creation activities can benefit from a dialogue with a GPT and incorporating its feedback. **I expect that you will use AI and so scoring will assume a high quality submission as a minimum bar, make sure you improve upon AI's output.**
2. Write in your own words even when GPT can generate almost finished work. A typical workflow can be like this: have a dialog with the GPT to get ideas, clarify what you want by critiquing its output, review the output to ensure it aligns with what you need, refine through more dialog, and then write it in your own words. **The submitted write-up should feel as if it is written by someone in your team, otherwise you will be penalized for inappropriate AI usage.**
3. Whenever you use AI in an artefact, ensure that you declare it - purpose and process of usage (why was it used, how was it used, extent of use, etc.)