

Team Homepage.....	2
Nguyen Duy Anh.....	3
Lai Gia Khanh.....	4
Nguyen Duc Nhan.....	5
Bui Thai Anh.....	6
Working Agreement.....	7
Underperformance Provision.....	10
Meeting notes.....	12
2023-09-09 Meeting notes.....	14
2023-09-11 Meeting notes.....	15
2023-09-18 Meeting notes.....	16
2023-09-25 Meeting notes.....	17
2023-10-02 Meeting notes.....	18
2023-10-09 Meeting notes.....	19
2023-10-16 Meeting notes.....	20
2023-10-26 Meeting notes.....	21
2023-11-03 Meeting notes.....	22
2023-11-11 Meeting notes.....	23
2023-11-22 Meeting notes.....	24
Project plan.....	26
Roles and Responsibilities.....	28
Risk Assessment Matrix.....	29
1st Persona and Empathy Map.....	31
2nd Persona and Empathy map.....	34
Product requirements.....	36
Prototype.....	37
Team Health Monitor.....	40
ER Diagram review.....	43
Physical Database.....	45
Document on data creation and null values.....	48
Query command (Use cases).....	52
Performance (Indexes).....	59
Major-specific work - Software.....	62
Team Retrospective.....	75
Jira Timeline.....	78

Team Homepage

Welcome to tankcos20031

Our mission is to revolutionize sustainable job discovery by creating a robust backend system and developing a comprehensive real-time database. Through intuitive interfaces and innovative technology, we connect job seekers and employers, empowering them to contribute to a better future.

<h4> Team metrics</h4> <p>Cohesion ON TRACK / OFF TRACK</p> <hr/> <p>Quality ON TRACK / OFF TRACK</p> <hr/> <p>Productivity ON TRACK / OFF TRACK</p>	<h4> About tankcos20031</h4> <p>We are a team of 4 friends. Our goal in this project is to deliver a prototype for The GreeLiving database project and achieve high marks in the semester.</p>	<h4> Deliverables</h4> <p><input checked="" type="checkbox"/> Project Proposal <input checked="" type="checkbox"/> Progress Report <input checked="" type="checkbox"/> Process and Product Video <input checked="" type="checkbox"/> Reflection and Peer Assessment <input checked="" type="checkbox"/> Final Report and Product Deliverable</p>				
<h4> Meet the team</h4> <table border="1"><tbody><tr><td><p>@Nguyen Duy Anh (Swinburne HN) Nguyen Duy Anh</p></td><td><p>@Nguyen Duc Nhan (Swinburne HN) Nguyen Duc Nhan</p></td></tr><tr><td><p>@Lai Gia Khanh (Swinburne HN) Lai Gia Khanh</p></td><td><p>@Bui Thai Anh (Swinburne HN) Bui Thai Anh</p></td></tr></tbody></table>			 <p>@Nguyen Duy Anh (Swinburne HN) Nguyen Duy Anh</p>	 <p>@Nguyen Duc Nhan (Swinburne HN) Nguyen Duc Nhan</p>	 <p>@Lai Gia Khanh (Swinburne HN) Lai Gia Khanh</p>	 <p>@Bui Thai Anh (Swinburne HN) Bui Thai Anh</p>
 <p>@Nguyen Duy Anh (Swinburne HN) Nguyen Duy Anh</p>	 <p>@Nguyen Duc Nhan (Swinburne HN) Nguyen Duc Nhan</p>					
 <p>@Lai Gia Khanh (Swinburne HN) Lai Gia Khanh</p>	 <p>@Bui Thai Anh (Swinburne HN) Bui Thai Anh</p>					

Contact

Nguyen Duy Anh
<ul style="list-style-type: none">Email: anhndswh00992@fpt.edu.vnTel: 0934646004

Nguyen Duc Nhan
<ul style="list-style-type: none">Email: nhanndswh01001@fpt.edu.vnTel: 0332360636

Lai Gia Khanh
<ul style="list-style-type: none">Email: khanhlgswh01380@fpt.edu.vnTel: 0903482358

Bui Thai Anh
<ul style="list-style-type: none">Email: anhbtswh00979@fpt.edu.vnTel: 0918988972

🎮 Nguyen Duy Anh

Student ID	<ul style="list-style-type: none">• Australia ID: 104212294• Viet Nam ID: SWH00992
Role	<ul style="list-style-type: none">• Project manager
Strength	<ul style="list-style-type: none">• Managing work
Contact	<ul style="list-style-type: none">• Email: anhndswh00992@fpt.edu.vn• Tel: 0934646004
Preferred Communication	<ul style="list-style-type: none">• Email• Messager

ℹ️ For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

⑧ Lai Gia Khanh

Student ID	<ul style="list-style-type: none">• Australia ID: 104222015• Viet Nam ID: SWH01380
Role	<ul style="list-style-type: none">• Developer
Strength	<ul style="list-style-type: none">• Coding
Contact	<ul style="list-style-type: none">• Email: khanhlgswh01380@fpt.edu.vn• Tel: 0903482358
Preferred Communication	<ul style="list-style-type: none">• Messenger, Zalo

 For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

Student ID	<ul style="list-style-type: none">• Australia ID: 104180485• Viet Nam ID: SWH01001
Role	<ul style="list-style-type: none">• Technical reporter/Developer
Strength	<ul style="list-style-type: none">• Working on documents and report• Basic coding
Contact	<ul style="list-style-type: none">• Email: nhanndswh01001@fpt.edu.vn• Tel: 0332360636
Preferred Communication	<ul style="list-style-type: none">• Messenger• Zalo• Email

 For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

 Bui Thai Anh

Student ID	<ul style="list-style-type: none">• Australia ID: 104221643• Viet Nam ID: SWH00979
Role	<ul style="list-style-type: none">• Developer
Strength	<ul style="list-style-type: none">• Basic coding
Contact	<ul style="list-style-type: none">• Email: anhbtswh00979@fpt.edu.vn• Tel: 0918988972
Preferred Communication	<ul style="list-style-type: none">• Online: Discord, Messenger, Google Meets• Offline: Swinburne's lab, cafeteria, team's working office

 For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

Working Agreement

Team Preferences

Team Member	 Bui Thai Anh h	 Nguyen Duy An h	 Nguyen Duc Nh an	 Lai Gia Khan h
Working location and timezone	Ha Noi, ICT	Ha Noi, ICT	Ha Noi, ICT	Ha Noi, ICT
Working hours and commitments	<ul style="list-style-type: none"> 8 - 12 am every Monday and Wednesday morning 8 - 9 pm every Tuesday, Thursday and Friday evenings 	<ul style="list-style-type: none"> 8-12 am every Monday morning 8-10 pm every Thursday and Friday 	<ul style="list-style-type: none"> Morning: Monday, Thursday Evening: from Tuesday to Saturday 	<ul style="list-style-type: none"> Morning: Monday, Wednesday, Thursday, Friday. Evening: from Tuesday to Saturday
Working environment and preferences	<ul style="list-style-type: none"> Swinburne's lab Library Cafeteria 	<ul style="list-style-type: none"> Office space Tuesday and Wednesday for other works 	<ul style="list-style-type: none"> At home Class 	<ul style="list-style-type: none"> Class Cafeteria
How I like receiving feedback	<ul style="list-style-type: none"> Giving feedbacks directly during meetings Discuss about my personal issue through social media platforms or emails 	<ul style="list-style-type: none"> Directly when discussing Through message and email 	<ul style="list-style-type: none"> Exchanging feedbacks directly during meetings Discuss other minor issues online 	<ul style="list-style-type: none"> Through message or email Directly in team meetings
Context about me	<ul style="list-style-type: none"> Be passionate about music and sports Be interested in movies, especially sci-fi and horror films 	<ul style="list-style-type: none"> Like reading 	<ul style="list-style-type: none"> Love watching and playing sports 	<ul style="list-style-type: none"> Love playing football

Communication Channels

Channel	Purpose	Audience	Standards
Email	Daily work, project outcome, milestone	Team member, stake holders	<ul style="list-style-type: none"> • Formal written documents • Keep emails concise and to the point • Attach necessary documents or links
Online chat	Feedback and discussion	Team members	<ul style="list-style-type: none"> • Use clear and concise language • Keep conversations focused on the topic
Face-to-face conversations	Feedback and discussion, report	Team member, stake holders	<ul style="list-style-type: none"> • Prepare an agenda or key talking points • Use visual aids for reports, if necessary • Follow up with written documentation when needed

📅 Meetings

Objective	Stakeholders update on project progress	Address roadblocks faced by team members	Regular team meeting
Outcomes	Stakeholders understand the project adn help the team with roadblock	Identify the problems and helping each others with the problems, ensure the project progress and planned	Progress report, roadblock, discussion about the project
Format	Whole team meeting with stakeholders via online meeting app or face-to-face	Whole team meeting via online meeting app or face-to-face	Whole team meeting via online meeting app or face-to-face
Who	Team members: @Nguyen Duy Anh (Swinburne HN) @Bui Thai Anh (Swinburne HN)	Team members: @Nguyen Duy Anh (Swinburne HN) @Bui Thai Anh (Swinburne HN)	Team members: @Nguyen Duy Anh (Swinburne HN) @Bui Thai Anh (Swinburne HN)

	@Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN) and stakeholders	@Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN)	@Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN)
Resources	Zoom, laptop, documents, camera, micro	Zoom, laptop, documents, camera, micro	Zoom, laptop, documents, camera, micro
How will we show up?	take responsibility for each of our work. We open up for vulnerability and create a safe working environment for everyone. Create and follow meeting to ensure that time is used efficiently	Open mindset, prepare to address the problems.	Prepare materials beforehand, active, compare work progress to the plan, address any problems or any additional ideas
How will we manage follow up?	Clear action items, dealines and responsible individuals. Meeting will be schedule for progress check.	Roadblocks will be documented, progress on resolving will be reviewed.	Each meeting will be doccumented.

⬆ Escalation Process

Decider	How	Transparency	Feedback Loop
Project Manager		The decision process will be shared between the team. Each team member can have input.	We will share the decision with stakeholders via emails or Atlassian

💡 Continuous Improvement

Purpose	How	Standards
Sharing feedbacks	Meet call, face-to-face meeting	Feedbacks will be noted
Milestone	Milestone	Online meeting will be recorded, milestone will be noted and compared to plan

Underperformance Provision

Purpose

This provision is designed to support and facilitate the improvement of team members who may be underperforming or contributing less than expected in a team project.

Process Overview

- Identification of Underperformance:** Team members should actively observe their peers during team meetings, collaborative discussions, and task assignments. Indicators of underperformance may include consistently missing deadlines, showing a lack of engagement, or not actively participating in team activities.

Poor Contribution Weighting Rubric			
0 % of Team mark (student receives no marks for team deliverable)	25% of Team Mark	50% of Team Mark	75% of Team Mark
May have attended some Class or Team Meetings. However, team member did not contribute meaningfully to team product and/or process AND/OR Team member has very limited understanding of team work product and/or process.	Team member made some attempt to contribute. However, behaviour of team member reduced team effectiveness, efficiency and morale.	Team member made some attempt to contribute. However, behaviour of team member reduced team on two of the following: effectiveness, efficiency and morale.	Team member made some attempt to contribute. However, behaviour of team member reduced team on one of the following: effectiveness, efficiency and morale.

- Initial Assessment:** The team should assess the situation and determine if the underperformance is affecting the overall team's progress or outcomes.
- Discussion and Communication:** Team members are encouraged to openly discuss concerns with the underperforming student to understand their perspective and challenges.
- Performance Improvement Plan (PIP):** If the underperformance continues and is deemed significant, a Performance Improvement Plan (PIP) may be developed collaboratively. The PIP should outline specific improvement goals, timelines, and support mechanisms.
- Faculty Involvement:** If necessary, the faculty facilitator or convener may be engaged to mediate discussions, provide guidance, or assess the impact of the underperformance on the team.
- Marking Adjustment:** In cases where the underperformance persists and affects team deliverables, marking adjustments may be considered by the convener based on evidence presented by the student, team, and/or facilitator.

Support Mechanisms:

- The team should aim to support the underperforming student by providing resources, guidance, or assistance.
- Faculty facilitators may offer guidance and mentoring to help the student meet expectations.
- The team should encourage open communication and collaboration to address challenges collectively.

Objective: The objective of this provision is to promote a collaborative approach to address teammate underperformance, provide opportunities for improvement, and ensure the equitable assessment of team deliverables.

Note: This provision is intended to be a flexible framework to support students in the event of underperformance. The focus is on constructive communication and improvement rather than punitive measures.

Meeting notes

[Create meeting note](#)

Incomplete tasks from meetings

Task report

Looking good, no incomplete tasks.

Decisions from meetings

Page Title	Decisions
2023-09-25 Meeting notes	 @... will submit the Project Proposal  TANK-49 - Submit the Project Proposal DONE
2023-10-02 Meeting notes	 Write SQL Commands to create different tables
2023-10-16 Meeting notes	 Complete all listed work and @... will submit the Progress Report to Canvas.
2023-10-26 Meeting notes	 Complete all tasks in  TANK-13 - Frontend DONE
2023-11-03 Meeting notes	 Improve and complete constructing the database
2023-11-11 Meeting notes	 Try to complete as many back-end functions as possible
2023-11-22 Meeting notes	 Complete all tasks in this final run-up and submit to Canvas before the due date

All meeting notes

Title	Creator	Modified
2023-11-11 Meeting notes	Nguyen Duc Nhan (Swinburne HN)	about an hour ago
2023-11-22 Meeting notes	Nguyen Duc Nhan (Swinburne HN)	about 14 hours ago
2023-11-03 Meeting notes	Nguyen Duc Nhan (Swinburne HN)	Nov 24, 2023
2023-10-26 Meeting notes	Nguyen Duc Nhan (Swinburne HN)	Nov 23, 2023
2023-10-09 Meeting notes	Nguyen Duy Anh (Swinburne HN)	Nov 23, 2023
2023-10-02 Meeting notes	Nguyen Duy Anh (Swinburne HN)	Nov 23, 2023
2023-09-25 Meeting notes	Nguyen Duc Nhan (Swinburne HN)	Nov 23, 2023
2023-10-16 Meeting notes	Nguyen Duy Anh (Swinburne HN)	Nov 23, 2023

2023-09-11 Meeting notes	Nguyen Duy Anh (Swinburne HN)	Nov 23, 2023
2023-09-18 Meeting notes	Nguyen Duy Anh (Swinburne HN)	Sept 30, 2023
2023-09-09 Meeting notes	Bui Thai Anh (Swinburne HN)	Sept 18, 2023

2023-09-09 Meeting notes

Date

9 Sept 2023

Participants

- @Bui Thai Anh (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)
- @Nguyen Duc Nhan (Swinburne HN)
- @Nguyen Duy Anh (Swinburne HN)

Goals

- Forming team
- Distributing workload
- Finish team agreement

Discussion topics

Time	Item	Presenter	Notes
10 am	Form team		
10:15 am	Working agreement	Duy Anh	
10:45 am	Profile		<ul style="list-style-type: none">• Each member finish their own profile for Homepage

Action items

- Form team
- Working agreement
- Profile

Decisions



2023-09-11 Meeting notes

Date

11 Sept 2023

Participants

- @Nguyen Duy Anh (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duc Nhan (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)

Goals

- Working on project proposal

Discussion topics

Time	Item	Presenter	Notes
10:00 am	Risk Assessment	@Nguyen Duc Nhan (Swinburne HN)	TANK-5: Risk Assessment DONE
11:00 am	Provision for teammate underperformance	@Nguyen Duy Anh (Swinburne HN)	TANK-64: Provision for teammate underperformance DONE

Action items

- Risk Assessment
- Provision for teammate underperformance

Decisions



2023-09-18 Meeting notes

Date

18 Sept 2023

Participants

- @Nguyen Duy Anh (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duc Nhan (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)

Goals

Working with Stakeholder

Discussion topics

Time	Item	Presenter	Notes
9 am - 10:15 am	Working with stakeholder via Google Meet	@Bui Thai Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN)	

Action items

- Take notes of the stakeholder's desires and requirements
- Ask about some unclear ideas of the project

Decisions



2023-09-25 Meeting notes

Date

25 Sept 2023

Participants

- @Nguyen Duc Nhan (Swinburne HN)
- @Nguyen Duy Anh (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)

Goals

- Working on the ER diagram
- Working on incomplete tasks of Project Proposal

Discussion topics

Time	Item	Presenter	Notes
10:20 am	ER diagram	@Bui Thai Anh (Swinburne HN)	Finish by Wednesday

Action items

- Ask and refer to different sources to complete the tasks as soon as possible
- Complete all undone tasks in TANK-1: Project Proposal DONE

Decisions

👉 @Nguyen Duc Nhan (Swinburne HN) will submit the Project Proposal
 TANK-49: Submit the Project Proposal DONE

2023-10-02 Meeting notes

Date

2 Oct 2023

Participants

- @Nguyen Duy Anh (Swinburne HN)
- @Nguyen Duc Nhan (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)

Goals

- Begin creating database

Discussion topics

Time	Item	Presenter	Notes
10:20	Creating database	@Bui Thai Anh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	Begin creating database based on initial ER diagram

Action items

- Planning tables
- Planning SQL Commands to create these tables
- TANK-9: Initial database DONE

Decisions

- ✓ Write SQL Commands to create different tables

2023-10-09 Meeting notes

Date

9 Oct 2023

Participants

- @Nguyen Duy Anh (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duc Nhan (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)

Goals

- Refining database

Discussion topics

Time	Item	Presenter	Notes
10:20	Refining database	@Bui Thai Anh (Swinburne HN)	After creating the database, double check with tutor to fix and optimize based on 3NF ➡ TANK-17: Updated ER diagram DONE

Action items

- ✓ Refining database
- ✓ Continue to do tasks in ➡ TANK-51: Enhanced database DONE

Decisions



2023-10-16 Meeting notes

Date

16 Oct 2023

Participants

- @Nguyen Duy Anh (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duc Nhan (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)

Goals

- Fixing Product Requirement
- Redoing Project Plan
- Discussing on Team Health Checks

Discussion topics

Time	Item	Presenter	Notes
10:00 AM	Fixing Product Requirement	@Lai Gia Khanh (Swinburne HN)	<ul style="list-style-type: none">• Connection to Personas• Redo scope, milestone• ▣ TANK-21: Redo Product Requirement DONE
11:00 AM	Redoing Project Plan	@Nguyen Duc Nhan (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	<ul style="list-style-type: none">• Timeline• Key Outcome• ▣ TANK-20: Fixing Project Plan DONE
9:00 PM	Discussing on Team Health Checks	All members	<ul style="list-style-type: none">• @Nguyen Duc Nhan (Swinburne HN) will take notes and complete the Team Health Monitor• ▣ TANK-22: Team Health Monitor DONE

Action items

- Edit the necessary fields in Product Requirement and Project Plan
- Planning to take actions on the focus area in Team Health Monitor

Decisions

👉 Complete all listed work and @Nguyen Duc Nhan (Swinburne HN) will submit the Progress Report to Canvas.

2023-10-26 Meeting notes

Date

26 Oct 2023

Participants

- @Nguyen Duc Nhan (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duy Anh (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)

Goals

- Develop front-end functions for the website

Discussion topics

Item	Presenter	Notes
Header/Footer	@Nguyen Duc Nhan (Swinburne HN)	Complete the frame for the website as soon as possible TANK-29: Header/Footer DONE
User registration	@Nguyen Duc Nhan (Swinburne HN)	TANK-14: User registration DONE
Login	@Nguyen Duc Nhan (Swinburne HN)	TANK-28: Login DONE
Home page	@Lai Gia Khanh (Swinburne HN)	TANK-35: Home page DONE
Course selection	@Lai Gia Khanh (Swinburne HN)	TANK-15: Course selection DONE
Job selection	@Nguyen Duc Nhan (Swinburne HN)	TANK-19: Job selection DONE

Action items

- @Lai Gia Khanh (Swinburne HN) and @Nguyen Duc Nhan (Swinburne HN) are responsible for using HTML and CSS to design all listed pages

Decisions

- 👉 Complete all tasks in TANK-13: Frontend DONE

2023-11-03 Meeting notes

Date

3 Nov 2023

Participants

- @Nguyen Duc Nhan (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duy Anh (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)

Goals

- Enhance the database
- Start working on back-end functions for the website

Discussion topics

Item	Presenter	Notes
Create and insert dummy data	@Bui Thai Anh (Swinburne HN)	TANK-46: Insert dummy data DONE
SQL use cases	@Nguyen Duy Anh (Swinburne HN)	TANK-53: Use cases and SQL statements, transactions DONE
Indexes	@Bui Thai Anh (Swinburne HN)	Try to improve performance by using indexes TANK-52: Performance (indexes) DONE
Planning for back-end functions	@Bui Thai Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN)	<ul style="list-style-type: none">• Discuss and summarize all necessary tasks for the back-end of the website• Add tasks on Jira

Action items

- @Nguyen Duy Anh (Swinburne HN) and @Bui Thai Anh (Swinburne HN) are responsible for doing database-related items in discussion
- @Nguyen Duc Nhan (Swinburne HN) is responsible for adding back-end tasks on Jira
- @Lai Gia Khanh (Swinburne HN) should quickly complete the missed deadline of creating front-end for home page [TANK-35: Home page](#) DONE

Decisions

- Improve and complete constructing the database

2023-11-11 Meeting notes

Date

11 Nov 2023

Participants

- @Nguyen Duc Nhan (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duy Anh (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)

Goals

- Develop the back-end for the website

Discussion topics

Item	Presenter	Notes
User registration	@Nguyen Duy Anh (Swinburne HN)	TANK-30: User registration DONE
Login	@Nguyen Duy Anh (Swinburne HN)	Try to do if possible
Register courses for clients	@Bui Thai Anh (Swinburne HN)	TANK-32: Register courses DONE
Apply jobs for clients	@Bui Thai Anh (Swinburne HN)	TANK-33: Apply jobs DONE
Client profile update	@Bui Thai Anh (Swinburne HN)	TANK-34: Profile update DONE

Action items

- @Bui Thai Anh (Swinburne HN) needs to complete the missed deadline of inserting dummy data DONE [TANK-46: Insert dummy data](#)
- @Bui Thai Anh (Swinburne HN) will attempt to make the back-end using PHP and JS. Other members will assist where needed

Decisions

👉 Try to complete as many back-end functions as possible

2023-11-22 Meeting notes

Date

22 Nov 2023

Participants

- @Nguyen Duc Nhan (Swinburne HN)
- @Bui Thai Anh (Swinburne HN)
- @Nguyen Duy Anh (Swinburne HN)
- @Lai Gia Khanh (Swinburne HN)

Goals

Complete:

- Process and Product slides and video
- Final Report and Product Deliverable assignment

Discussion topics

Item	Presenter	Notes
+ TANK-26: Process and Product Video DONE	All members	<p>Requirements:</p> <p>3 minute video on the process and product created for this project comprising of:</p> <ol style="list-style-type: none">1. Title slide with team names;2. Introduction and project background, summary of requirements;3. Mission statement and objectives;4. 1-2 slides about team process (collaboration, roles, contributions to solution);5. 1-3 slides about the logical/physical design, discussion of decisions made regarding the entities and their relationships;6. Discussion of solutions to use cases (e.g. SQL, also the design solutions influenced by these use cases - there is overlap with item 5 here);7. 1-5 slides on major-specific technology8. Summary of solution and lessons learnt.
+ TANK-25: Final Report and Product Deliverable DONE	All members	<p>Requirements:</p> <ul style="list-style-type: none">• Initial ER diagram (Week 4)• Review of ER diagram to normalise/denormalise, revise relationships (Week 5), possible subsequent reviews (e.g. adjustments due to use cases)• Physical database (Week 6) - Create Table statements• Document on data creation and null values (Week 7)• Use cases and SQL statements, transactions (Week 8)

- Performance (indexes) (Week 9)
- Major-specific work (**naming the contributor**) (Week 10 + 11)
- Team reflection (Week 12)
- Completed Jira Roadmap

Action items

- All team members work on the tasks as assigned on Jira
- Try to complete all work on time

Decisions

 Complete all tasks in this final run-up and submit to Canvas before the due date

Project plan

Driver	Instructors
Approver	Instructors
Contributors	@Nguyen Duy Anh (Swinburne HN) @Bui Thai Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN) and stakeholders
Informed	Greeliving Learning Hub
Objective	A dynamic website with proper backend and database to process users' requests for Greeliving Learning Hub.
Due date	27 Nov 2023
Key outcomes	<ul style="list-style-type: none"> Design and implement a project plan using industry standard project management and collaboration tools. Effectively discuss project-related considerations and deliverables with stakeholders, using industry-standard tools. Contribute to the project development as a respectful and responsible team member. A full fledge dynamic website include frontend, backend and a database to handle user's requests and fits with GreeLiving hub requirement
Status	DONE

🧠 Problem Statement

Problem: GreeLiving currently lacks a solid backend system with an integrated Content administration System (CMS) and real-time database administration. This flaw reduces the platform's ability to provide a consistent user experience. Users struggle to navigate the platform and obtain relevant educational programs and job ads, limiting their capacity to seek worldwide education and work possibilities.

Impact: The lack of an efficient CMS and real-time database system has a negative impact on user experiences. They face difficulties in locating appropriate courses, obtaining credible information, and achieving job placements, both locally and globally. This problem not only impedes individual development but also GreeLiving's aim of empowering users through lifelong learning and professional development.

Hypothesis: We believe that by building a complete backend system with a user-friendly CMS and incorporating a dynamic database management system, we can considerably improve the GreeLiving platform. We predict that this upgrade will result in the following outcomes:

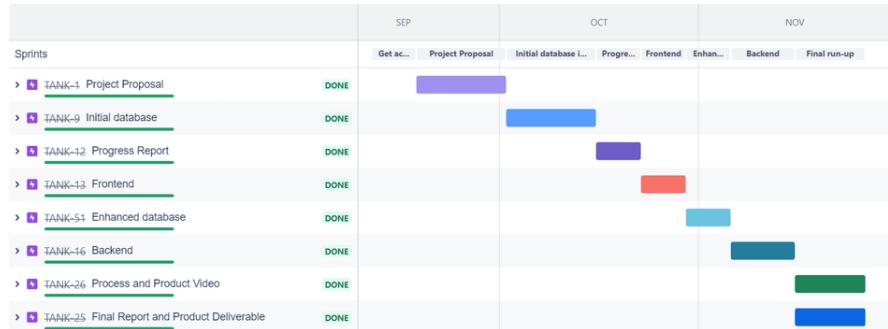
- Improved User Experience:** Users will navigate the platform with ease, accessing the information and opportunities they need.
- Efficient Content Management:** The CMS will streamline content updates and site management.
- Real-time Job Access:** The database system will provide up-to-date job listings and candidate profiles.

🎯 Scope

Must have:	<ul style="list-style-type: none"> Database: The project will comprise the creation and deployment of a strong database management system. This system will make it easier to store and retrieve real-time data, such as job ads and candidate profiles. The database will comprise of several table to store users informations, courses informations, Job informations.... The database will conform to the 3NF.
Nice to have:	<ul style="list-style-type: none"> Website: A website or app prototype, while not required, may be considered as part of the project. This prototype can be used to visualize the platform's user interface and capabilities. The frontend will be made using HTML and CSS while the backend will be made utilising PHP along with Javascript to interact with the database.
Not in scope:	<ul style="list-style-type: none"> Content development: This project does not include the creation of instructional content, job ads, or candidate profiles. The emphasis is on technical infrastructure.

📅 Timeline

[https://tankcos20031.atlassian.net/jira/software/projects/TANK/boards/2/timeline?
shared=&at!Origin=eyJpIjoiOWUzZWFiMWY0ZDFhNDRmYjk4MTM2M2YmMwMmNiOWUiLCJwljoiaiJ9](https://tankcos20031.atlassian.net/jira/software/projects/TANK/boards/2/timeline?shared=&at!Origin=eyJpIjoiOWUzZWFiMWY0ZDFhNDRmYjk4MTM2M2YmMwMmNiOWUiLCJwljoiaiJ9)



▶ Milestones and deadlines

Milestone	Owner	Deadline	Status
Project Proposal <ul style="list-style-type: none"> • Team homepage • Meeting notes • Working agreement • Project plan • Roles and Responsibilities • Risk Assessment • Persona • Empathy Map • Product requirement • Underperformance Provision 	@Bui Thai Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	1 Oct 2023	DONE
Initial database <ul style="list-style-type: none"> • ER diagram (3NF) • MySQL script 	@Bui Thai Anh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	15 Oct 2023	DONE
Prototype <ul style="list-style-type: none"> • Frontend • Backend • Database 	@Bui Thai Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	15 Nov 2023	IN PROGRESS

Roles and Responsibilities

📋 Overview

Identify and discuss team responsibilities by following the instructions for the [Roles and Responsibilities Play](#).

Team	Team TANK
Team members	@Nguyen Duy Anh (Swinburne HN) @Bui Thai Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN)
Date	19 Sept 2023
Team mission	Develop a comprehensive database and prototype for GreeLiving that enable customers to register for course and seek job opportunities.

📘 Roles and responsibilities

Roles	Responsibilities (what others think)	Responsibilities (what I think)	Assigned
Project manager	<ul style="list-style-type: none">• Lead training workshop• Arrange team meeting	<ul style="list-style-type: none">• Discussion with stakeholders• Receive feedback and relay back to team members• Solving unexpected problems	@Nguyen Duy Anh (Swinburne HN)
Developer	<ul style="list-style-type: none">• Coding• Create websites or apps	<ul style="list-style-type: none">• Create a finished product (websites, apps) based on the prototype to solve stakeholders' issues	@Lai Gia Khanh (Swinburne HN) @Bui Thai Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN)
Business analyst/reporter	<ul style="list-style-type: none">• Gather documents' requirements• Problem identification• Figuring out strategies for the process improvement	<ul style="list-style-type: none">• Analyzing information using some analytical tools and techniques• Communicate with stakeholders and developers• Create the connection between developer team and customers	@Nguyen Duc Nhan (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)

Risk Assessment Matrix

Background

- Enhance an existing website that serves as a platform for individuals searching for part-time job opportunities and seeking to acquire relevant skills through courses
- Develop and implement a database management system to use live data for job listings and candidate profiles
- Develop a backend system for managing the website by various users
- Recognize possible obstacles to succeed in executing a project
- Develop mitigation strategies and response plans for the identified risks

Risks management

	Risks	Mitigation strategies	Risk rating
			Severity (1-4)
			Likelihood (1-3)
1	Data breaches and user data privacy violations	Implement robust security measures, follow security best practices; comply with relevant data protection regulations and implement privacy policies	High 4 1
2	Software bugs and compatibility issues	Conduct thorough testing and quality assurance processes throughout the development lifecycle; use automated testing tools, perform compatibility testing across different browsers and devices; consult with teammates and instructors when needed	Medium 2 3
3	Server failure or downtime	Implement a reliable hosting provider with backup systems and monitoring tools to ensure server stability and minimize downtime.	Medium 3 2
4	Economic downturn or changes in industry demand	Diversify the courses and job possibilities available to meet the needs of various companies and sectors; keep up to date on economic trends and industry projections; maintain the ability to change offerings in response to changing market conditions	Medium 2 2
5	Fraudulent job postings or scams	Implement a robust job posting and employer verification process.; inform users about potential job scams and provide guidance for identifying and reporting suspicious activities; regularly monitor job postings and respond to any detected fraudulent activities	Medium 3 1
6	Negative reviews or feedback from learners or job seekers	Create a mechanism for gathering and responding to customers' feedback as soon as possible; address concerns and issues raised by learners or job seekers transparently and professionally	Medium 2 2

7	Poor communication and coordination among team members	Establish open communication ways and hold regular team meetings; encourage open and transparent communication in order to address any problems or concerns promptly	High 3 1
8	Differences in work styles or conflicting priorities	Encourage open dialogue and foster a collaborative team environment; clearly outline roles, responsibilities, and project dates to match team members' expectations; create a method for resolving issues and foster a culture of collaboration and mutual respect	Medium 3 2
9	Underperformance	Define performance expectations clearly and communicate them to team members; provide regular feedbacks and assistance to assist them in enhancing their performance; address issues and obstacles promptly	High 3 2
10	Project delays or missed deadlines	Create a project timeline that includes clear milestones and deadlines; monitor development on a regular basis and identify any obstacles; implement time management techniques	High 3 1

Risk rating

LOW	MEDIUM	HIGH	EXTREME
<ul style="list-style-type: none"> Acceptable Ok to proceed 	<ul style="list-style-type: none"> As low as reasonably practicable Take mitigation efforts 	<ul style="list-style-type: none"> Generally unacceptable Seek support 	<ul style="list-style-type: none"> Intolerable Place event on hold

LIKELIHOOD	SEVERITY			
	1- ACCEPTABLE <i>Little to no effect on event</i>	2 - TOLERABLE <i>Effects are felt, but not critical to outcome</i>	3 - UNDESIRABLE <i>Serious impact to the course of action and outcome</i>	4 - INTOLERABLE <i>Could result in disaster</i>
1- IMPROBABLE <i>Risk is unlikely to occur</i>			5; 7; 10	1
2- POSSIBLE <i>Risk will likely occur</i>		2; 4; 6	8; 9	
3- PROBABLE <i>Risk will occur</i>	3			

1st Persona and Empathy Map



Persona name	Neil Mayor
Persona role	Junior web developer
Job description	Developing frontend

🏢 Recent company

Company name	Lambdabet
Company size	~20000 employees
Industry	IT

👤 Demographic information

Age	23
Gender	Male
Income	\$50000 per year
Education level	Bachelor degree in Computer Science
Residential environment	Urban

✍️ Personal quote

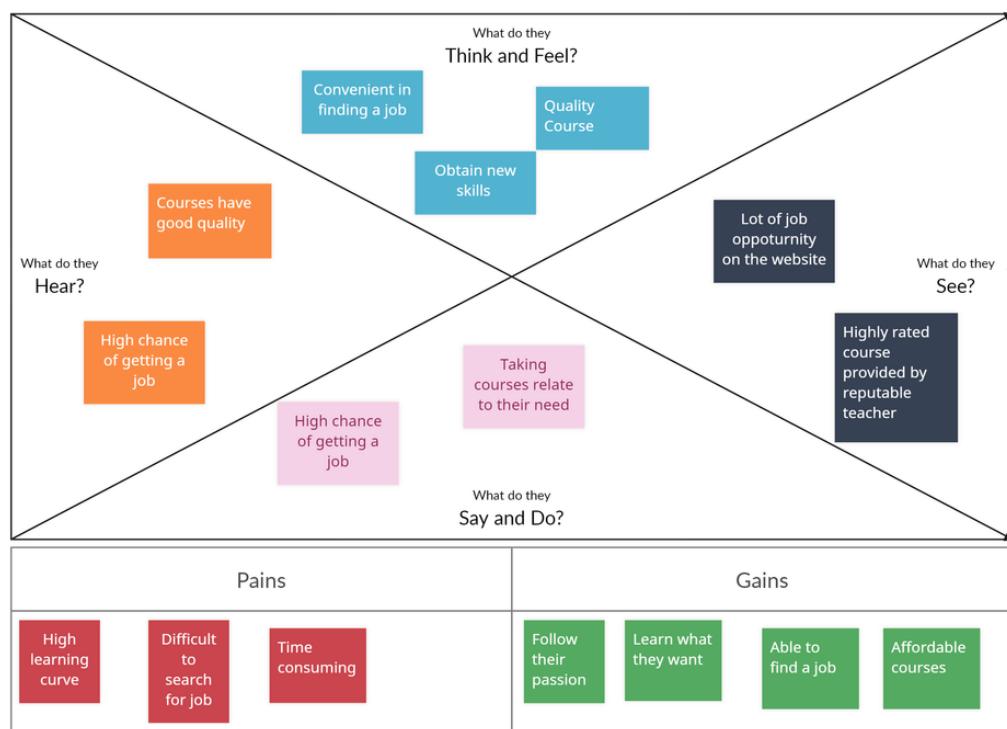
"Life's a journey, not a destination. Embrace the unknown, learn from the glitches, and keep upgrading yourself along the way."

📋 Biography

From a young age, Neil has always been fascinated with computers. He pursued an IT career, joining a computer science course with the ambition to develop a website that could change the world, similar to Facebook. However, after graduating from college, he discovered that his true passion lies in the field of AI.

Professional goals	Motivators
<ul style="list-style-type: none"> Neil wants to learn about Machine Learning and AI to later become an ML/AI architecture. 	<ul style="list-style-type: none"> Neil was inspired by the work of his professor at college who is an ML/AI architecture at a reputable company, and he also thinks that in the future, AI is going to change the world in unimaginable ways.
Challenges	Sources of information
<ul style="list-style-type: none"> He only has experience and expertise in web development 	<ul style="list-style-type: none"> N/A

Empathy Map



2nd Persona and Empathy map



Persona name	David Beckham
Persona role	Client
Job description	The president and co-owner of Inter Miami FC

🏢 Company

Company name	Inter Miami FC
Industry	Football/soccer

👤 Demographic information

Age	48
Gender	Male
Income	\$40 million - \$50 million
Education level	High school

✍️ Personal quote

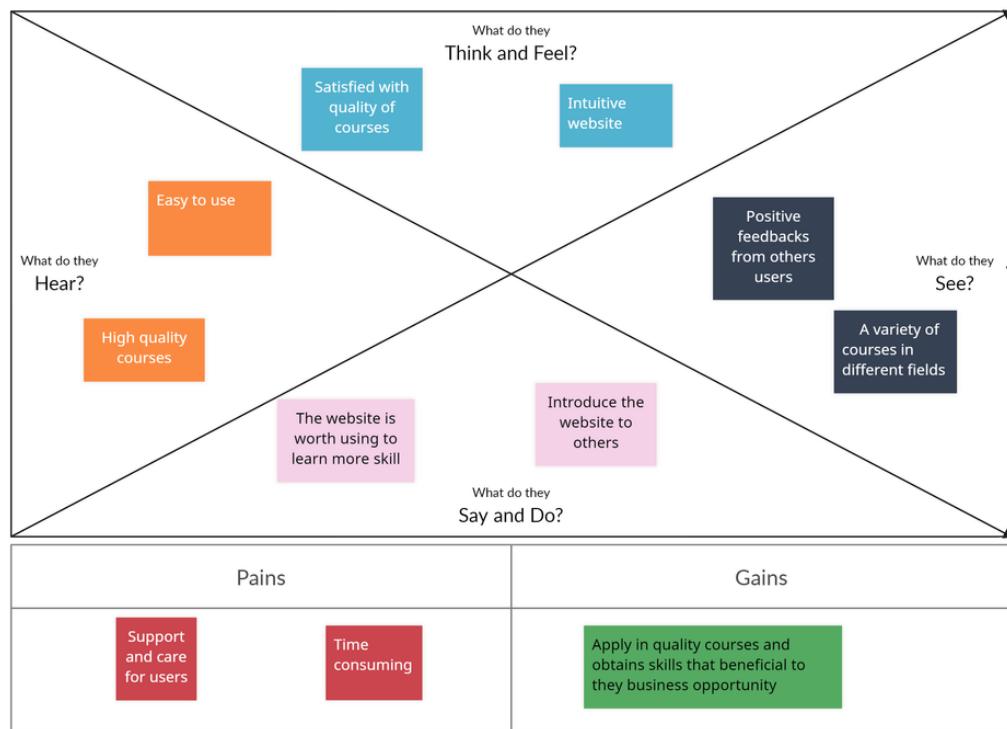
"I never do anything half-heartedly. I will continue to work hard and play hard and do everything I can to be successful, whatever I do."

Biography

David Beckham, born on May 2, 1975, in London, England, had a storied football career. He began at Manchester United's youth academy and rose to prominence for his precise crosses and free-kicks. Beckham won numerous titles with Manchester United and formed a legendary partnership with the "Class of '92." In 2003, he made a high-profile move to Real Madrid, helping them secure a La Liga title. He then played for the LA Galaxy, contributing to the growth of soccer in the US. Beckham also represented England, earning 115 caps and participating in several international tournaments. After retiring in 2013, he pursued business ventures, becoming a fashion icon, owning a football team, and supporting UNICEF, solidifying his status as a football legend on and off the field.

Professional goals	Motivators
<ul style="list-style-type: none"> Beckham wants to expand his business to other sports such as basketball, gymnastic, e-sports,.... 	<ul style="list-style-type: none"> Want to earn more money and develop his brand
Challenges	Sources of information
<ul style="list-style-type: none"> Large workload because he is the president of a football club Not good with tech in general 	<ul style="list-style-type: none"> Read information on the Internet

Empathy Map



Product requirements

[Add Product requirements](#)

Shown in the Prototype page below

Error while fetching page properties report data: Bad Request

Prototype

Target release	26 Nov 2023
Epic	+ TANK-13: Frontend DONE + TANK-51: Enhanced database DONE + TANK-16: Backend DONE
Document status	DRAFT
Document owner	@Nguyen Duy Anh (Swinburne HN)
Designer	@Lai Gia Khanh (Swinburne HN)
Tech lead	@Bui Thai Anh (Swinburne HN)
Technical reporter	@Nguyen Duc Nhan (Swinburne HN)

🎯 Objective

The project is a strategic initiative aimed to promoting lifelong learning, expanding global reach, enhancing workforce development, embracing digital transformation, and prioritizing user-centricity. It aligns directly with GreeLiving's mission to empower individuals through education and career opportunities.

The main goal of the project is to create a dynamic website which is a seamless recruiting platform to connect candidates and employers.

Some objectives are aimed to achieve:

- Database Design: create the database structure, optimize data storage and ensure efficient data retrieval.
- Content Management System: create a system for managing the website by different users.
- Web Security: secure the website against online threats like hacking, data breaches and malware.
- E-Commerce Development: create online stores (shopping cart,...) and payment systems (payment gateways and user authentication).
- UI/UX Design: design user-friendly and visually appealing.
- Search Engine Optimization (SEO): improve the visibility in search engine results.
- Responsive Website Design: make sure that the website adapt and function well on different screen size, devices and popular Operating Systems such as Windows, Linux,...

📊 Success metrics

Goal	Metric
Simplify the user experience	Customer satisfaction score
Increasing User Engagement	Daily active user
Facilitate Job Placement	Job Placement Success Rate

🤔 Assumptions

The GreeLiving initiative's assumptions for our project team include the premise that our consumers have basic digital literacy abilities and ready access to digital tools and internet connectivity. We think that our users are eager to learn more about overseas educational and employment prospects. On the technical side, we acknowledge data security, scalability, and cross-device compatibility as limits. Furthermore, we foresee significant market demand, potential for successful relationships with educational institutions and employers, and

many revenue creation paths. To ensure the project's eventual success, we must validate these assumptions through extensive investigation and analysis.

🌟 Milestones

Task	Date	Description
 Database Structure	15/10	Define data schema and relationships.
 Frontend Magic	29/10	Implement responsive, user-centric design.
 Backend	12/11	Develop data processing and logic.
 Final Prototype	12/11	Integration and testing for the takeoff.

📝 Requirements

Requirement	User Story	Importance
Mobile responsive	Beckham has to be on the road a lot due to his work, therefore, he wants to be able to access GreeLiving platform on his mobile device so that he can browse courses anytime and anywhere.	MEDIUM
User Feedback Mechanism	Tuan found an error while using the product but could not report it.	HIGH
User Job Application	For Neil, he wants to easily create account, upload his CV and other documents and apply for job after a few clicks or have a detailed instructions during the experience on the website. He also wants to keep track of his recruitment process.	HIGH
User friendly	Beckham has little to none tech experince in general and in website like Greeliving, he wishes to have a straightforward experience, easy to navigate throughout the website.	HIGH
Management System	As a project manager, I want to keep track of the project progress, stay informed and identify any problems of the website.	HIGH

?

Open Questions

Question	Answer	Date Answered
How do we notify users of update	We will notify them using in-app message and also through email for important update	26 Sept 2023

Scope

- Dynamic website, users can put in their information like age, gender, etc...
- The website will have a course webpage where all course provided by GreeLiving will be displayed and users can choose which course to enroll, they will be able to filter courses as well.
- Jobs' webpage display job opportunities from different employers. It will display requirements from employee, salary, location, etc...
- Database structure to store all the information about users, jobs and courses.

⚠ Out of Scope

- Multilingual support
- Web Analytics
- Internationalization and Localization

Team Health Monitor

Team name	TANK
Sponsor	Greeliving
Health monitor cadence	once every 2 weeks

TEAM health assessment

With your team, read the definition of each attribute of healthy, high-performing teams out loud. On the count of three have each person rate how they feel the team is doing compared to each definition (thumbs-up/green, thumbs-sideways/yellow, thumbs-down/red). Record the results of each attribute rating in the table. Highlight each cell using this color code: **HEALTHY** = "We're strong here", **BIT SICK** = "We're ok... but a little shaky", **SICK** = "We're not healthy".

Area	Date 16 Oct 2023
 Full-time owner There is one lead who is accountable for the result of this project. This needs to be someone whose time is at least 80% dedicated to it, and who can champion the mission inside and outside of the team.	Duy Anh is our full-time owner, however, he finds it almost impossible to dedicate 80% of his time to this project. Anyway, it does not seem like a major issue to us.
 Balanced team Roles and responsibilities are clear and agreed upon. The project has people with the right blend of skill set. Acknowledge that team members can change by stage.	We have agreed upon the main responsibilities of each member, but when we are in the progress of doing the project, we may have to make several modifications in order to adapt to the actual situation.
 Shared understanding The team has a common understanding of why they're here , the problem/need, are convinced about the idea, confident they have what they need, and trust each other.	Everyone on the team has a common understanding of the project's purpose, problem/need, and idea. We all know what we are working on and why we are doing this project.
 Value and metrics It's clear what success means from a business and user's perspective, and there is a unique value proposition in place for the target users and to the business. Success is defined, with a goal, and how it will be measured.	We have defined measures of success by setting our own goals quite clearly. Our progress is tracked via Jira and it is checked on a regular basis to make sure that all the targets are achievable.

Proof of concept	Some sort of demonstration has been created and tested, that demonstrates why this problem needs to be solved, and demonstrates its value.	Although our team has engaged in extensive discussions regarding the project, we have yet to create a tangible demonstration that will help visualize our current progress.
One-pager	The project is summarized in a one-pager and shared with anyone so that they understand the purpose of the project, and its value.	Our one-pager has already been shown on the Product requirement/Prototype page on our Confluence space.
Managed dependencies	Clear understanding of complexity, infrastructure involved, risks, resources, effort, and timeline. Clear understanding of who we depend on, and who depends on us.	By and large, we have mostly comprehended the project requirements. On the contrary, we need to further communicate with stakeholders involved in our project and establish a clear understanding of the dependencies in order to effectively manage them
Velocity	The team is making incremental progress by shipping concrete iterations to stakeholders (and, even better, to production), learning along the way, and implementing lessons learned , resulting in greater success.	We are making progress, nevertheless, we should contribute more time to be able to achieve the targets of our project.

🎯 Focus areas

Ask your team to collectively come up with one attribute you want to focus on. Then, call out ways to move the **SICK** or **BIT SICK** toward **HEALTHY**. Make sure they are actionable, specific, and measurable.

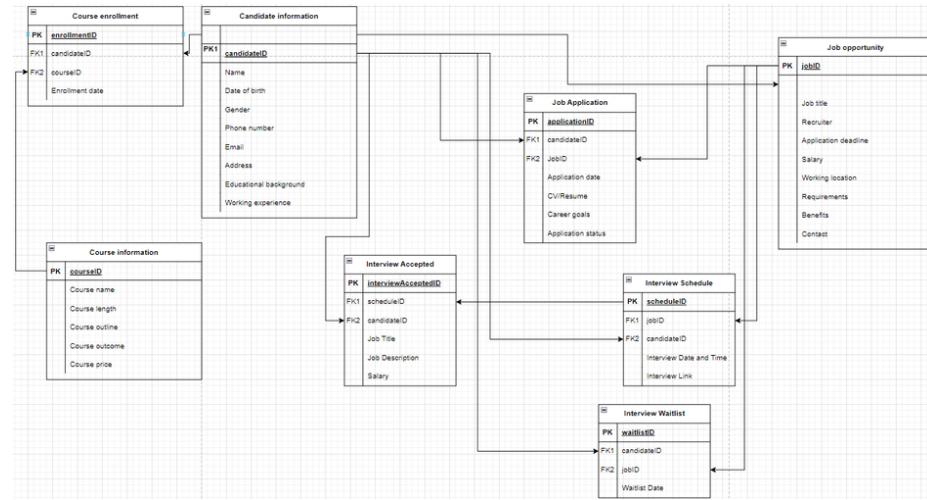
Date	Focus areas and action items
16 Oct 2023	<p>Focus area: Proof of concept</p> <p>Action items:</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Define the purpose and scope of the demonstration <input checked="" type="checkbox"/> Break down the demonstration creation process into smaller tasks <input checked="" type="checkbox"/> Assign responsibility for each task <input checked="" type="checkbox"/> Set a deadline for creating the demonstration

Next steps

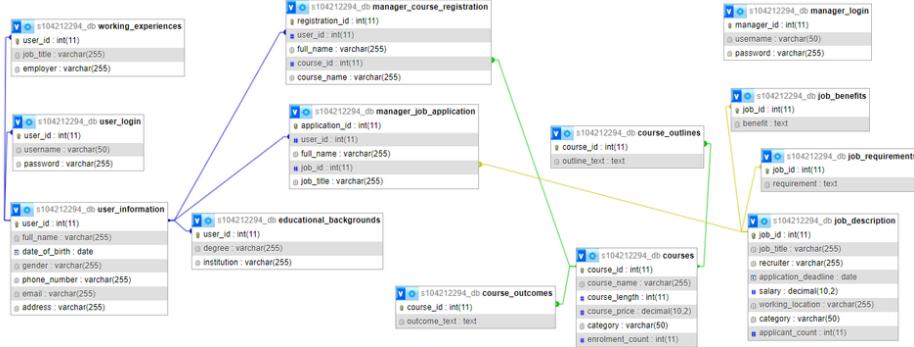
Consider running the plays in suggested in [Step 2 of the facilitation instructions](#) for improving **BIT SICK** and **SICK** attribute areas. Don't treat these as prescriptions! You know your team better than anyone, so check them out, [explore other plays](#), and do what you think is best.

ER Diagram review

Initial ER Diagram:



Final version:



Overall, our ER diagram has conformed to 3NF but still need improvement in some places:

Relationship:

- The relationship between 'user_information' and 'user_login', 'educational_background' and 'work_experiences' is one-to-one relationship since one user can only have one profile. Those table share 'user_id' as primary key and foreign key for 'educational_background', 'work_experiences' and 'user_login'. 'user_login' are separated as they are credential of the user and need to separate for security reason.
- The relationship between 'courses' and 'course_outcomes', 'course_outlines' is one-to-one relationship . Those table share 'course_id' as primary key and foreign key for 'course_outcomes' and 'course_outlines'
- The relationship between 'job_description' and 'job_benefits', 'job_requirements' is one-to-one relationship . Those table share 'job_id' as primary key and foreign key for 'job_benefits' and 'job_requirements'

User Information Table:

- Key:** user_id (Primary Key)

◦ **Explanation:** The user_id serves as the primary key, uniquely identifying each user in the table. This conforms to 3NF as it ensures that each record is uniquely identified by a primary key.

Educational Backgrounds and Working Experiences Tables:

- Key:** user_id (Primary Key)

◦ **Explanation:** The user_id in these tables is the primary key, ensuring that each educational background and working experience is uniquely associated with a user. The foreign key relationship with user_information(user_id) further establishes a link between these tables, maintaining 3NF principles.

User Login Table:

- Key:** user_id (Primary Key)

◦ **Explanation:** The user_id serves as the primary key, uniquely identifying each user login entry. The foreign key relationship with user_information(user_id) ensures that each login entry corresponds to a valid user, adhering to 3NF.

Job Description Table:

- **Key:** `job_id` (Primary Key)
 - **Explanation:** The `job_id` acts as the primary key, uniquely identifying each job description. This conforms to 3NF, ensuring unique identification of records in the table.

Job Requirements and Job Benefits Tables:

- **Key:** `job_id` (Primary Key)
 - **Explanation:** The `job_id` is the primary key in these tables, uniquely identifying each set of job requirements and benefits. The foreign key relationship with `job_description(job_id)` maintains 3NF by associating requirements and benefits with specific job descriptions.

Courses, Course Outlines, and Course Outcomes Tables:

- **Key:** `course_id` (Primary Key)
 - **Explanation:** The `course_id` serves as the primary key in these tables, ensuring unique identification of each course, outline, and outcome. The foreign keys (`course_id` in `course_outlines` and `course_id` in `course_outcomes`) maintain 3NF by establishing relationships between the tables.

Manager Course Registration and Manager Job Application Tables:

- **Keys:** `registration_id` and `application_id` (Primary Keys)
 - **Explanation:** Both tables have primary keys (`registration_id` and `application_id`) ensuring unique identification of each registration and application. The foreign keys (`user_id`, `course_id`, and `job_id`) establish relationships with other tables (`user_information`, `courses`, and `job_description`), adhering to 3NF.

Manager Login Table:

- **Key:** `manager_id` (Primary Key)
 - **Explanation:** The `manager_id` acts as the primary key, uniquely identifying each manager's login information. This conforms to 3NF as it ensures unique identification of login records.

Overall:

- In summary, the primary keys in each table uniquely identify records, conforming to 3NF.
- Foreign keys are appropriately defined, establishing relationships between tables and maintaining referential integrity.
- The structure of the database, in terms of keys, aligns with the principles of the Third Normal Form.

Compared to the first version of the ER diagram, this final version is better-fitted to 3NF. The 'Educational background' and 'Work experience' was a part of 'Candidate Information', furthermore, those attributes didn't specify which information to be stored, which can lead to double value. The same is applied for job's requirements and benefits. Course's outline and outcome also face the same problem. In the final version, this has been improved. For instance, the 'Work experience' and 'Educational background' has been separated into different table: 'work_experiences' and 'educational_background'. In each table, different attributes are created and specified in the web that user can only enter one information:

```
CREATE TABLE educational_backgrounds (
    user_id INT NOT NULL,
    degree VARCHAR(255) NOT NULL,
    institution VARCHAR(255) NOT NULL,
    PRIMARY KEY (user_id),
    FOREIGN KEY (user_id) REFERENCES user_information(user_id)
);

CREATE TABLE working_experiences (
    user_id INT NOT NULL,
    job_title VARCHAR(255) NOT NULL,
    employer VARCHAR(255) NOT NULL,
    PRIMARY KEY (user_id),
    FOREIGN KEY (user_id) REFERENCES user_information(user_id)
);
```

New attributes include 'degree', 'institution', 'job_title' and 'employer'. This is to make sure there is no double value.

Physical Database



```
1  CREATE TABLE user_information (
2      user_id INT NOT NULL AUTO_INCREMENT,
3      full_name VARCHAR(255) NOT NULL,
4      date_of_birth DATE NOT NULL,
5      gender VARCHAR(255) NOT NULL,
6      phone_number VARCHAR(255) NOT NULL,
7      email VARCHAR(255) NOT NULL,
8      address VARCHAR(255) NOT NULL,
9      PRIMARY KEY (user_id)
10 );
11
12 CREATE TABLE educational_backgrounds (
13     user_id INT NOT NULL,
14     degree VARCHAR(255) NOT NULL,
15     institution VARCHAR(255) NOT NULL,
16     PRIMARY KEY (user_id),
17     FOREIGN KEY (user_id) REFERENCES user_information(user_id)
18 );
19
20 CREATE TABLE working_experiences (
21     user_id INT NOT NULL,
22     job_title VARCHAR(255) NOT NULL,
23     employer VARCHAR(255) NOT NULL,
24     PRIMARY KEY (user_id),
25     FOREIGN KEY (user_id) REFERENCES user_information(user_id)
26 );
27
28 CREATE TABLE user_login (
29     user_id INT,
30     username VARCHAR(50) NOT NULL,
31     password VARCHAR(255) NOT NULL,
32     PRIMARY KEY (user_id),
33     FOREIGN KEY (user_id) REFERENCES user_information (user_id)
34 );
35
36
37 CREATE TABLE job_description (
38     job_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
39     job_title VARCHAR(255) NOT NULL,
40     recruiter VARCHAR(255) NOT NULL,
41     application_deadline DATE NOT NULL,
42     salary DECIMAL(10,2) NOT NULL,
43     working_location VARCHAR(255) NOT NULL,
44     category VARCHAR(50) NOT NULL,
45     applicant_count INT NOT NULL DEFAULT 0
```

```

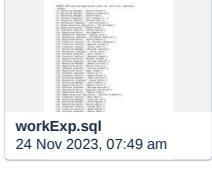
46 );
47
48 CREATE TABLE job_requirements (
49   job_id INT NOT NULL,
50   requirement TEXT NOT NULL,
51   PRIMARY KEY (job_id),
52   FOREIGN KEY (job_id) REFERENCES job_description(job_id)
53 );
54
55 CREATE TABLE job_benefits (
56   job_id INT NOT NULL,
57   benefit TEXT NOT NULL,
58   PRIMARY KEY (job_id),
59   FOREIGN KEY (job_id) REFERENCES job_description(job_id)
60 );
61
62
63 CREATE TABLE courses (
64   course_id INT NOT NULL AUTO_INCREMENT,
65   course_name VARCHAR(255) NOT NULL,
66   course_length INT NOT NULL,
67   course_price DECIMAL(10,2) NOT NULL,
68   category VARCHAR(50) NOT NULL,
69   enrolment_count INT NOT NULL DEFAULT 0,
70   PRIMARY KEY (course_id)
71 );
72
73 CREATE TABLE course_outlines (
74   course_id INT NOT NULL,
75   outline_text TEXT NOT NULL,
76   PRIMARY KEY (course_id),
77   FOREIGN KEY (course_id) REFERENCES courses(course_id)
78 );
79
80 CREATE TABLE course_outcomes (
81   course_id INT NOT NULL,
82   outcome_text TEXT NOT NULL,
83   PRIMARY KEY (course_id),
84   FOREIGN KEY (course_id) REFERENCES courses(course_id)
85 );
86
87 CREATE TABLE manager_course_registration (
88   registration_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
89   user_id INT NOT NULL,
90   full_name VARCHAR(255) NOT NULL,
91   course_id INT NOT NULL,
92   course_name VARCHAR(255) NOT NULL,
93   FOREIGN KEY (user_id) REFERENCES user_information(user_id),
94   FOREIGN KEY (course_id) REFERENCES courses(course_id)
95 );
96
97
98 CREATE TABLE manager_job_application (
99   application_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
100  user_id INT NOT NULL,
101  full_name VARCHAR(255) NOT NULL,
102  job_id INT NOT NULL,
103  job_title VARCHAR(255) NOT NULL,

```

```
104     FOREIGN KEY (user_id) REFERENCES user_information(user_id),
105     FOREIGN KEY (job_id) REFERENCES job_description(job_id)
106 );
107
108
109
110 CREATE TABLE manager_login (
111     manager_id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
112     username VARCHAR(50) NOT NULL,
113     password VARCHAR(255) NOT NULL
114 );
```

Document on data creation and null values

Full source code can be accessed via this link: [GitHub - tankcos20031/Database](#)

 courses.sql 24 Nov 2023, 07:49 am	Generate course
 Preview unavailable courseOutline.sql 24 Nov 2023, 07:49 am	Generate course outline
 Preview unavailable courseOutcome.sql 24 Nov 2023, 07:49 am	Generate course outcome
 Preview unavailable userInfo.sql 24 Nov 2023, 07:49 am	Generate user informations
 Preview unavailable eduBackground.sql 24 Nov 2023, 07:49 am	Generate user education background
 workExp.sql 24 Nov 2023, 07:49 am	Generate user past work experiences
 Preview unavailable jobDescription.sql 24 Nov 2023, 07:50 am	Generate job description
 Preview unavailable	Generate job benefits

<p>Preview unavailable</p> <p>jobBenefits.sql</p> <p>24 Nov 2023, 07:50 am</p>	
<p> Preview unavailable</p> <p>jobRequire.sql</p> <p>24 Nov 2023, 08:07 am</p>	<p>Generate job requirement (file is too large so i trimmed it to fit in this table, full file on github)</p>

- Firstly, we used the Faker library and random library in the Python ecosystem to create much more dummy data we wanted.
 - Secondly, we created some variables as tuples to store many random relevant data of the variables' names.
 - Lastly, we used a FOR loop to create much more data as we expected.

```
userinputpy | x
D:\> userinputpy >-
1  from faker import Faker
2
3  fake = Faker()
4  #dateofbirth
5  birth = ['15/07/1985', '23/04/1992', '08/11/1977', '14/02/1989', '30/09/1995', '05/06/1982', '19/12/1973', '02/03/1988', '11/10/1999', '27/08/1974', '06/01/1990', '22/06/1978', '09/0
6
7  #address
8  address = ['123 Main Street Anytown USA', '456 Elm Avenue, Suburbia, Canada', '789 Oak Lane, Villageville, UK', '101 Pine Road, Hamlet City, Australia', '202 Maple Street, Townsville
9  '123 Main Street, Tokyo, Japan', '456 Elm Avenue, Seoul, South Korea', '789 Oak Drive, Beijing, China', '321 Pine Road, Mumbai, India', '654 Cedar Lane, Jakarta, Indonesia', '987 Ma
10
11 print("INSERT INTO user_information (full_name, date_of_birth, gender, phone_number, email, address)n VALUES")
12
13 for i in range(500000):
14     name = fake.name()
15     dateOfBirth = fake.random_element(elements=birth)
16     gender = fake.random_element(elements=['Male', 'Female'])
17     email = fake.email()
18     phone_number = fake.phone_number()
19     userAddress = fake.random_element(elements=address)
20
21     print(f"({name}), ({dateOfBirth}), ({gender}), ({phone_number}), ({email}), ({userAddress})")
22
23 if i == 599999:
24     print(f"({name}), ({dateOfBirth}), ({gender}), ({phone_number}), ({email}), ({userAddress})");
25     break
```

Eg: This is a Python file for creating 600,000 rows of dummy data for the 'user_information' table in our database

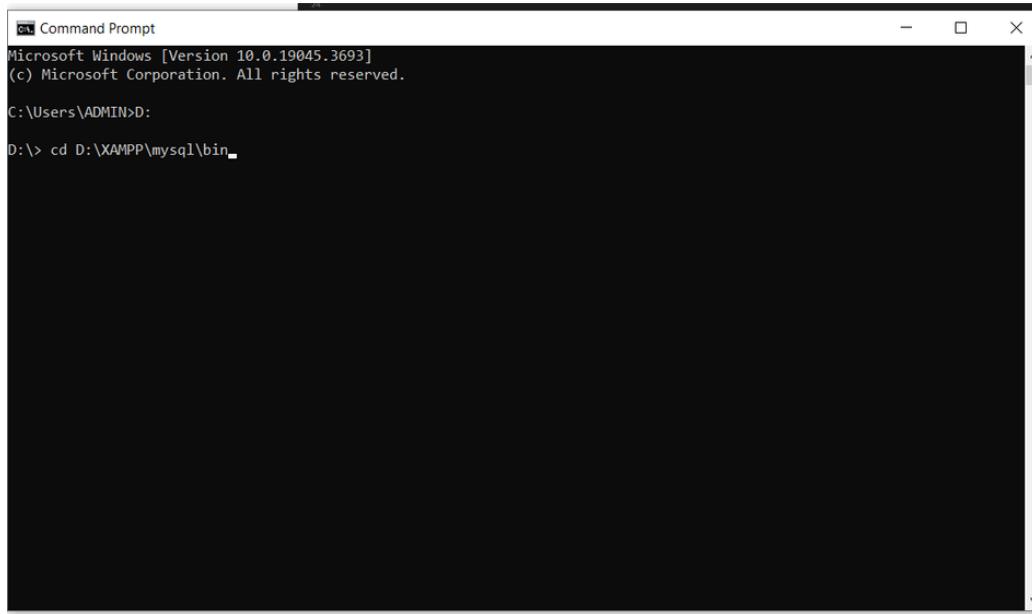
However, we occasionally have problems creating random data because of duplicate information or a lack of unique data. This happened due to the large amount of pre-trained data in variables. As a result, we decided to add other libraries, such as random, to assist us avoid having duplicate data.

```
courses.py x
D:\> coursespy >_
 1 #import random
 2 from faker import Faker
 3
 4 fake = Faker()
 5
 6 #course name
 7 courseName = ['Computer Science', 'Data Science', 'Web Development', 'Cybersecurity',
 8 'Digital Marketing', 'Business Administration', 'Finance', 'Marketing', 'Project Management', 'Public Relations', 'Film Production', 'Graphic Design', 'Civil Engin
 9 'Mechanical Engineering', 'Electrical Engineering', 'Chemical Engineering',
10 'Health Sciences', 'Nursing', 'Pharmacy', 'Public Health', 'Medicine', 'Physical Therapy', 'Education', 'Early Childhood Education', 'Curriculum and Instructio
11 'Nutrition and Dietetics', 'Fitness and Wellness', 'Health Informatics', 'Teaching English as a Second Language (TESOL)', 'Linguistics and Language Education']
12
13 #course length
14 courseLength = ['10 weeks', '11 weeks', '12 weeks', '13 weeks', '14 weeks', '15 weeks', '16 weeks', '17 weeks', '18 weeks', '19 weeks', '20 weeks', '21 weeks',
15 '22 weeks', '23 weeks', '24 weeks', '25 weeks', '26 weeks', '27 weeks', '28 weeks', '29 weeks', '30 weeks', '31 weeks', '32 weeks', '33 weeks', '34 weeks', '35
16 '36 weeks', '37 weeks', '38 weeks', '39 weeks', '40 weeks']
17 #course price
18 coursePrice = ['152', '91', '267', '122', '75', '198', '105', '284', '68', '160', '234', '54', '117', '88', '203', '99', '77', '116', '215', '59', '270',
19 '176', '83', '124', '189', '72', '150', '195', '106', '63', '78', '282', '131', '59', '101', '271', '219', '113', '65', '88', '157', '285', '209', '79',
20 '273', '97', '252', '182', '54', '120', '66', '194', '282', '84']
21
22 #category
23 major = 'IT', 'Business', 'Media', 'Health', 'Education', 'Engineering'
24
25 #enrolment count
26 count = '5', '10', '15', '20', '25', '30', '35', '40', '45', '50'
27
28 print("INSERT INTO courses (course_name, course_length, course_price, category, enrolment_count)\n VALUES")
29 for i in range(600000):
30
31     course_name = fake.random.choice(courseName)
32     course_length = fake.random.choice(courseLength)
33     course_price = fake.random.choice(coursePrice)
34     category = fake.random_element(elements=(major))
35     enrolment_count = fake.random_element(elements=(count))
36
37     print("(" + course_name + ", " + course_length + ", " + course_price + ", " + category + ", " + enrolment_count + ")")
```

Eg: We decided to use a "random" library so we also had to modify code syntax to create much more unique dummy data for 'course_information' table in the database.

Overall, we decided to create approximately 600,000 dummy data, which is a pretty enormous data source. Although creating so much dummy data is very difficult for us at the first time since we did not find any tools to create a lot of fake data, we decided to make use of Python library. However, we figured out that we could not upload a large data file into feenix mariadb or phpmyadmin, thus we could only push this much data on localhost database, but we commit that our dummy data is good and redundant enough for us to both test our SQL functions and implementing indexing on database efficiently.

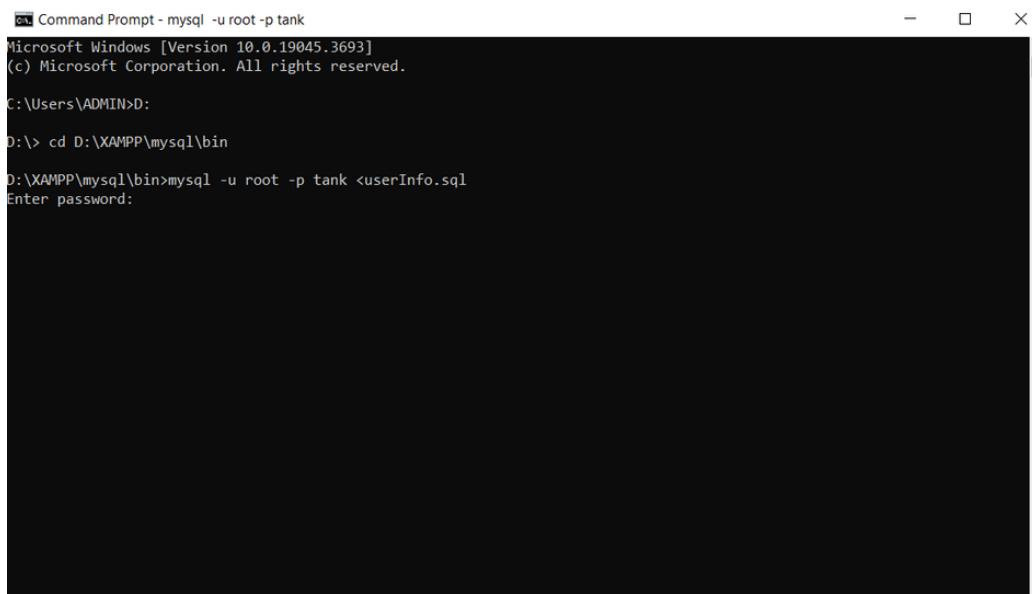
In terms of pushing this enormous data on localhost phpmysql, we cannot use the normal way since the system does not allow users to insert a large data file. Therefore, we decided to use command line to deal with this problem



```
Command Prompt
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>D:
D:\> cd D:\XAMPP\mysql\bin
```

Firstly, we redirect to the path above D:\XAMPP\mysql\bin.



```
Command Prompt - mysql -u root -p tank
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>D:
D:\> cd D:\XAMPP\mysql\bin

D:\XAMPP\mysql\bin>mysql -u root -p tank <userInfo.sql
Enter password:
```

After that we copy and paste a random data file into this file location in our personal laptop. For example in this case, we will push a data file called userInfo.sql on a database called tank. Finally, we use a syntax "mysql -u username -p database_name <(filename).sql". In this case, we used "mysql -u root -p tank <userInfo.sql". Since it is the localhost, we do not need any passwords here, just enter and wait for the system to insert that file into database.

phpMyAdmin

Server 127.0.0.1 > Database tank > Table user_information

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Showing rows 0 - 499 (595480 total, Query took 0.0016 seconds.)

SELECT * FROM `user_information`

Profiling | Edit inline | Edit | Explain SQL | Create PHP code | Refresh |

1 >> Show all Number of rows: 500 Filter rows: Search this table Sort by key: None

Extra options

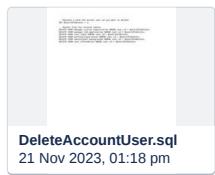
	user_id	full_name	date_of_birth	gender	phone_number	email	address
<input type="checkbox"/>	2	Juan Russo	27/08/1977	Female	(404)970-8301x88390	marshamy@example.org	4444 Juniper Avenue, Valleyville, Greece
<input type="checkbox"/>	3	Kevin Nolan	09/12/2001	Female	+1-529-741-7365x275	brandonhunter@example.org	3939 Sequoia Lane, Lakeshore, Switzerland
<input type="checkbox"/>	4	Logan Harrington	08/11/1977	Male	662-839-3034x33499	ronald48@example.net	2222 Birch Avenue, Riverside, Vietnam
<input type="checkbox"/>	5	Matthew Barnes	22/12/1984	Male	916-567-4408	sonyachaney@example.com	456 Elm Avenue, Suburbia, Canada
<input type="checkbox"/>	6	Richard Newton	07/06/1980	Male	+1-267-285-5331	bil29@example.com	808 Spruce Lane, Hilltop, Mexico
<input type="checkbox"/>	7	Stacey Pena	10/06/1990	Male	590 499.0651x70733	johnjohnson@example.net	101 Pine Road, Hamlet City, Australia
<input type="checkbox"/>	8	Harold Middleton	01/09/1998	Female	725-625-0104	jasonpeterson@example.com	4848 Redwood Street, Lakeside, Romania
<input type="checkbox"/>	9	Rebecca Hansen	08/02/2002	Female	725.772.0991	kaitlynholmes@example.net	1515 Maple Avenue, Highlands, South Korea
<input type="checkbox"/>	10	Kathleen Smith	03/07/2000	Female	321.503.6238x904	preynolds@example.com	1717 Willow Lane, Rivertown, Egypt

As you can see, we succeeded in inserting a large data file with approximately 600,000 records to the database.

Query command (Use cases)

Full source code can be accessed via this link: [GitHub - tankcos20031/Database](#)

File	Description
 InsertUserInfor.sql 21 Nov 2023, 01:09 pm	Insert user information at register page
 InsertCourses.sql 21 Nov 2023, 01:10 pm	Insert user information at register page
 InsertJobs.sql 21 Nov 2023, 01:10 pm	Insert new Job (can insert multiple jobs)
 RegisterCourseJob.sql 21 Nov 2023, 01:12 pm	User register for course and apply for job
 Filter.sql 21 Nov 2023, 01:12 pm	Filter query for job and course by categories and salary, price range
 ApplicantOfCo... Job.sql 21 Nov 2023, 01:16 pm	Retrieve information about applicants for a specific course, job
 CourseOfUser.sql 21 Nov 2023, 01:17 pm	Retrieve information about courses that a specific person has enrolled in

 JobApplyOfUser.sql 21 Nov 2023, 01:18 pm	Retrieve information about jobs that a specific person has applied for
 DeleteAccountUser.sql 21 Nov 2023, 01:18 pm	Delete user account
 LeaveCourseJob.sql 21 Nov 2023, 01:18 pm	Leave applied jobs and course

Insert user information at register page

```

1 -- Insert user information
2 INSERT INTO user_information (full_name, date_of_birth, gender, phone_number, email, address)
3 VALUES
4 ('John Doe', '1990-05-15', 'Male', '123-456-7890', 'john.doe@example.com', '123 Main St, Cityville');
5
6 -- Get the user_id generated for the inserted user information
7 SET @user_id := LAST_INSERT_ID();
8
9 -- Insert educational background
10 INSERT INTO educational_backgrounds (user_id, degree, institution)
11 VALUES
12 (@user_id, 'Bachelor of Science', 'University A');
13
14 -- Insert working experience
15 INSERT INTO working_experiences (user_id, job_title, employer)
16 VALUES
17 (@user_id, 'Software Engineer', 'Tech Company X');
18
19 -- Insert user login
20 INSERT INTO user_login (user_id, username, password)
21 VALUES
22 (@user_id, 'john_doe', 'password123');
```

Insert new Course (can insert multiple courses)

```
1 -- Insert course information
2 INSERT INTO courses (course_name, course_length, course_price, category, enrolment_count)
3 VALUES
4 ('Web Development Fundamentals', 8, 199.99, 'Programming', 0);
5
6 -- Get the course_id generated for the inserted course information
7 SET @course_id_1 := LAST_INSERT_ID();
8
9 -- Insert course outline
10 INSERT INTO course_outlines (course_id, outline_text)
11 VALUES
12 (@course_id_1, 'Introduction to HTML and CSS, Basic JavaScript Concepts, Building Responsive Websites');
13
14 -- Insert course outcome
15 INSERT INTO course_outcomes (course_id, outcome_text)
16 VALUES
17 (@course_id_1, 'Ability to create and style a basic website, Understanding of fundamental programming concepts')
18
19 -- Insert another course
20 INSERT INTO courses (course_name, course_length, course_price, category, enrolment_count)
21 VALUES
22 ('Data Science Bootcamp', 12, 299.99, 'Data Science', 0);
23
24 -- Get the course_id generated for the inserted course information
25 SET @course_id_2 := LAST_INSERT_ID();
26
27 -- Insert course outline for the second course
28 INSERT INTO course_outlines (course_id, outline_text)
29 VALUES
30 (@course_id_2, 'Introduction to Data Analysis, Machine Learning Fundamentals, Data Visualization Techniques');
31
32 -- Insert course outcome for the second course
33 INSERT INTO course_outcomes (course_id, outcome_text)
34 VALUES
35 (@course_id_2, 'Proficiency in data analysis tools, Understanding of machine learning concepts');
36
```

Insert new Job (can insert multiple jobs)

```
1 -- Insert job information
2 INSERT INTO job_description (job_title, recruiter, application_deadline, salary, working_location, category, app
3 VALUES
4 ('Software Engineer', 'Tech Company X', '2023-12-15', 80000.00, 'Cityville', 'IT', 0);
5
6 -- Get the job_id generated for the inserted job information
7 SET @job_id_1 := LAST_INSERT_ID();
8
9 -- Insert job requirements
10 INSERT INTO job_requirements (job_id, requirement)
11 VALUES
12 (@job_id_1, 'Bachelor's degree in Computer Science or related field');
```

```

14 -- Insert job benefits
15 INSERT INTO job_benefits (job_id, benefit)
16 VALUES
17 (@job_id_1, 'Health insurance, retirement plan, flexible working hours');
18
19 -- Insert another job
20 INSERT INTO job_description (job_title, recruiter, application_deadline, salary, working_location, category, app
21 VALUES
22 ('Data Scientist', 'Data Analytics Corp', '2023-12-25', 90000.00, 'Tech City', 'IT', 0);
23
24 -- Get the job_id generated for the inserted job information
25 SET @job_id_2 := LAST_INSERT_ID();
26
27 -- Insert job requirements for the second job
28 INSERT INTO job_requirements (job_id, requirement)
29 VALUES
30 (@job_id_2, 'Master's degree in Data Science or a related field');
31
32 -- Insert job benefits for the second job
33 INSERT INTO job_benefits (job_id, benefit)
34 VALUES
35 (@job_id_2, 'Flexible work schedule, stock options');
36

```

Insert into manager course table with update enrolment count

```

1 -- Assuming the user with user_id = 1 exists and wants to register for an existing course with course_id = 1
2 SET @user_id := 1;
3 SET @course_id := 1;
4
5 -- Fetch user's full_name based on user_id
6 SELECT full_name INTO @full_name FROM user_information WHERE user_id = @user_id;
7
8 -- Fetch course details based on course_id
9 SELECT course_name INTO @course_name FROM courses WHERE course_id = @course_id;
10
11 -- Insert the course registration record
12 INSERT INTO manager_course_registration (user_id, full_name, course_id, course_name)
13 VALUES (@user_id, @full_name, @course_id, @course_name);
14
15 -- Increment the enrolment_count in courses
16 UPDATE courses SET enrolment_count = enrolment_count + 1 WHERE course_id = @course_id;
17

```

Insert into manager job table with update applicant count

```

1 -- Assuming the user with user_id = 1 exists and wants to apply for an existing job with job_id = 1
2 SET @user_id := 1;
3 SET @job_id := 1;
4
5 -- Fetch user's full_name based on user_id
6 SELECT full_name INTO @full_name FROM user_information WHERE user_id = @user_id;

```

```

7
8 -- Fetch job details based on job_id
9 SELECT job_title INTO @job_title FROM job_description WHERE job_id = @job_id;
10
11 -- Insert the job application record
12 INSERT INTO manager_job_application (user_id, full_name, job_id, job_title)
13 VALUES (@user_id, @full_name, @job_id, @job_title);
14
15 -- Increment the applicant_count in job_description
16 UPDATE job_description SET applicant_count = applicant_count + 1 WHERE job_id = @job_id;
17

```

Filter queries

Filter Course by category

```

1 -- Replace 'YourCategory' with the actual category you want to filter
2 SELECT * FROM courses WHERE category = 'Programming';

```

Filter Course by price

```

1 -- Replace the values in the range with your actual price range
2 SELECT * FROM courses WHERE course_price BETWEEN 100.00 AND 500.00;

```

Filter Job by category

```

1 -- Replace 'YourCategory' with the actual category you want to filter
2 SELECT * FROM job_description WHERE category = 'IT';

```

Filter Job by price salary

```

1 -- Replace the values in the range with your actual price range
2 SELECT * FROM job_description WHERE salary BETWEEN 50000.00 AND 80000.00;

```

Retrieve information about applicants for a specific course

```

1 -- Replace 1 with the actual course_id you are interested in
2 SELECT
3     c.course_id,
4     c.course_name,
5     c.enrolment_count,
6     mcr.registration_id,
7     mcr.user_id,
8     ui.full_name
9 FROM
10    courses c
11 LEFT JOIN
12    manager_course_registration mcr ON c.course_id = mcr.course_id

```

```

13 LEFT JOIN
14     user_information ui ON mcr.user_id = ui.user_id
15 WHERE
16     c.course_id = 1; -- Replace 1 with the actual course_id you are interested in
17

```

Retrieve information about applicants for a specific job

```

1 -- Replace 1 with the actual job_id you are interested in
2 SELECT
3     jd.job_id,
4     jd.job_title,
5     jd.applicant_count,
6     mja.application_id,
7     mja.user_id,
8     ui.full_name
9 FROM
10    job_description jd
11 LEFT JOIN
12    manager_job_application mja ON jd.job_id = mja.job_id
13 LEFT JOIN
14    user_information ui ON mja.user_id = ui.user_id
15 WHERE
16     jd.job_id = 1; -- Replace 1 with the actual job_id you are interested in
17

```

Retrieve information about courses that a specific person has enrolled in

```

1 -- Replace 1 with the actual user_id you are interested in
2 SELECT
3     mcr.registration_id,
4     mcr.user_id,
5     ui.full_name,
6     c.course_id,
7     c.course_name
8 FROM
9     manager_course_registration mcr
10 JOIN
11    user_information ui ON mcr.user_id = ui.user_id
12 JOIN
13    courses c ON mcr.course_id = c.course_id
14 WHERE
15     mcr.user_id = 1; -- Replace 1 with the actual user_id you are interested in
16

```

Retrieve information about jobs that a specific person has applied for

```

1 -- Replace 1 with the actual user_id you are interested in
2 SELECT

```

```

3     mja.application_id,
4     mja.user_id,
5     ui.full_name,
6     jd.job_id,
7     jd.job_title
8 FROM
9     manager_job_application mja
10 JOIN
11     user_information ui ON mja.user_id = ui.user_id
12 JOIN
13     job_description jd ON mja.job_id = jd.job_id
14 WHERE
15     mja.user_id = 1; -- Replace 1 with the actual user_id you are interested in

```

Delete user account

```

1 -- Replace 1 with the actual user_id you want to delete
2 SET @userIdToDelete = 1;
3
4 -- Delete from the related tables
5 DELETE FROM manager_course_registration WHERE user_id = @userIdToDelete;
6 DELETE FROM manager_job_application WHERE user_id = @userIdToDelete;
7 DELETE FROM user_login WHERE user_id = @userIdToDelete;
8 DELETE FROM working_experiences WHERE user_id = @userIdToDelete;
9 DELETE FROM educational_backgrounds WHERE user_id = @userIdToDelete;
10 DELETE FROM user_information WHERE user_id = @userIdToDelete;

```

Leave course

```

1 -- Replace 1 with the actual registration_id of the course registration you want to delete
2 DELETE FROM manager_course_registration WHERE registration_id = 1;

```

Leave job apply

```

1 -- Replace 1 with the actual application_id of the job application you want to withdraw
2 DELETE FROM manager_job_application WHERE application_id = 1;

```

Performance (Indexes)

Based on the provided SQL statements, we did some research, analyzed each table, and identified potential columns that could benefit from indexing:

1. user_information table:

- We considered adding an index on the “email” column if you want to search for users by their email addresses, especially since it is really useful for a company to send specific announcements to each distinguished user
- To create the index, you can use the following SQL statement: CREATE INDEX idx_user_information_email ON user_information(email);
- For example, in this case, we want to find who has the email called “wandrews@example.org”

The screenshot shows a MySQL query results interface. At the top, a green bar indicates "Showing rows 0 - 2 (3 total, Query took 0.3149 seconds)". Below it is the SQL query: `SELECT * FROM `user_information` WHERE user_information.email = 'wandrews@example.org';`. The main area displays a table with three rows of data:

	user_id	full_name	date_of_birth	gender	phone_number	email	address
<input type="checkbox"/>	18	Walter Martinez	09/12/2001	Male	911.209.3626x2019	wandrews@example.org	1515 Maple Avenue, Highlands, South Korea
<input type="checkbox"/>	347890	Erika Matthews	16/01/1992	Male	828.806.6791	wandrews@example.org	4040 Spruce Road, Hillview, Austria
<input type="checkbox"/>	578441	Paige King	07/06/1980	Female	+1-958-881-7067x333	wandrews@example.org	1919 Sequoia Road, Hillside, Indonesia

At the bottom, there are buttons for "Check all", "With selected:", "Edit", "Copy", "Delete", and "Export".

After we use an index for the email column, it makes our query faster

The screenshot shows the same MySQL query results interface after an index has been created. The green bar at the top still says "Showing rows 0 - 2 (3 total, Query took 0.0006 seconds)". The SQL query remains the same: `SELECT * FROM `user_information` WHERE user_information.email = 'wandrews@example.org';`. The data table is identical to the previous one, showing three rows of user information.

We use the EXPLAIN syntax to check whether the index is used when running the query in this case. Fortunately, in this case, we can optimize the query by using the index.

The screenshot shows the MySQL EXPLAIN results for the query. The top bar says "Your SQL query has been executed successfully." The query is `EXPLAIN SELECT * FROM `user_information` WHERE user_information.email = 'wandrews@example.org';`. The results table shows the execution plan:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	user_information	ref	idx_user_information_email	idx_user_information_email	1022	const	3	Using index condition

2. educational_backgrounds table:

- The `user_id` column is already indexed due to the foreign key constraint.
- However, we prepared a lot of dummy data about university of users, so we decided to add an index in 'institution' column.
- For example, we want to find who has a Computer Science degree from Imperial College London

MySQL returned an empty result set (i.e. zero rows) (Query took 0.2034 seconds.)

```
SELECT * FROM `educational_backgrounds` WHERE educational_backgrounds.institution = 'Imperial College London' AND educational_backgrounds.degree = 'Computer science';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

user_id	degree	institution
---------	--------	-------------

Show results operations

After we used indexing, it made our query faster

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0005 seconds.)

```
SELECT * FROM `educational_backgrounds` WHERE educational_backgrounds.degree = 'Computer Science' AND educational_backgrounds.institution = 'Imperial College London';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

user_id	degree	institution
---------	--------	-------------

Show results operations

We also use EXPLAIN to check whether the index is used, it is successful to implement indexing in this case.

Your SQL query has been executed successfully.

```
EXPLAIN SELECT * FROM `educational_backgrounds` WHERE educational_backgrounds.institution = 'Imperial College London' AND educational_backgrounds.degree = 'Computer Sci'
```

[Edit inline] [Edit] [Skip Explain SQL] [Analyze Explain at mariadb.org] [Create PHP code]

Extra options

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	educational_backgrounds	ref	idx_educational_backgrounds_institution	idx_educational_backgrounds_institution	1022	const	1	Using index condition; Using where

3. working_experiences table:

- The `user_id` column is already indexed due to the foreign key constraint.

4. user_login table:

- The `username` column could be indexed for faster retrieval of user information based on the `username`.
- SQL syntax: `CREATE INDEX idx_user_login_username ON user_login(username);`

5. courses table:

- No potential columns for indexing. The primary key column (`course_id`) is already indexed by default.

6. course_outlines table:

- The `course_id` column is already indexed due to the foreign key constraint.

7. course_outcomes table:

- The `course_id` column is already indexed due to the foreign key constraint.

8. job_description table:

- We prepared so much dummy data about the category and salary of many jobs, so we decided to add indexes in 2 columns: `category` and `salary`.
- Query : Filter by Category and Salary at the same time :**
 - An index on the `category` column would be beneficial for this query as it involves a direct equality condition.
 - An index on the `salary` column could improve the performance of this query since it involves a range condition.

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0008 seconds.)

```
SELECT ui.user_id, ui.full_name FROM user_information ui JOIN manager_job_application mja ON ui.user_id = mja.user_id JOIN job_description jd ON mja.job_id = jd.job_id WHERE jd.category = 'YourSpecificCategory' AND jd.salary BETWEEN 50000 AND 80000;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

- After we used indexing, it worked properly and made the query faster

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0006 seconds.)

```
SELECT ui.user_id, ui.full_name FROM user_information ui JOIN manager_job_application mja ON ui.user_id = mja.user_id JOIN job_description jd ON mja.job_id = jd.job_id WHERE jd.category = 'IT' AND jd.salary BETWEEN 50000 AND 80000;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

user_id	full_name
---------	-----------

9. job_requirements table:

- The `job_id` column is already indexed due to the foreign key constraint.

10. `job_benefits` table:

- The `job_id` column is already indexed due to the foreign key constraint.

11. `manager_course_registration` table:

- The `registration_id` column is already a primary key and indexed.
- The `user_id` and `course_id` columns are already indexed due to the foreign key constraints.

12. `manager_job_application` table:

- The `application_id` column is already a primary key and indexed.
- The `user_id` and `job_id` columns are already indexed due to the foreign key constraints.

Major-specific work - Software

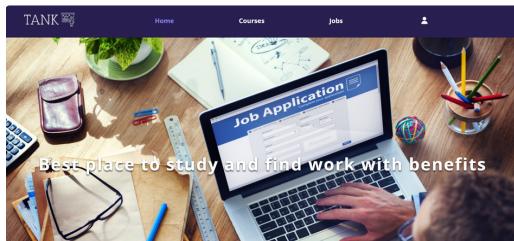
Full source code can be accessed via this link: [GitHub - tankcos20031/Database](https://github.com/tankcos20031/Database)

Set up instruction: included in the <https://github.com/tankcos20031/Database/blob/main/README.md> file on GitHub

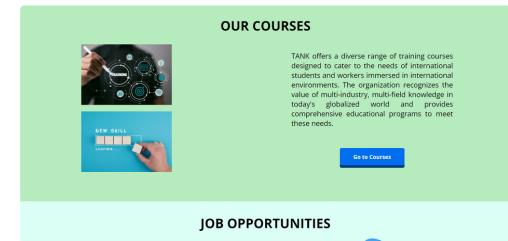
1. UI/UX Design: @Nguyen Duc Nhan (Swinburne HN) @Lai Gia Khanh (Swinburne HN)

Utilizing mainly HTML and CSS to create the front-end, we have crafted an intuitive and visually appealing user interface. The design concept includes a modern and clean style, as well as a carefully chosen color palette and font that corresponds to the website's purpose and target audience. The layout and navigation have been well organized to guarantee simple access to content and a smooth user experience.

Here are some website prototype images:



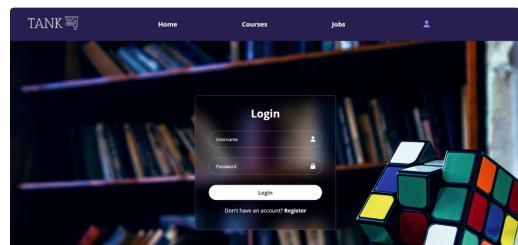
Home page



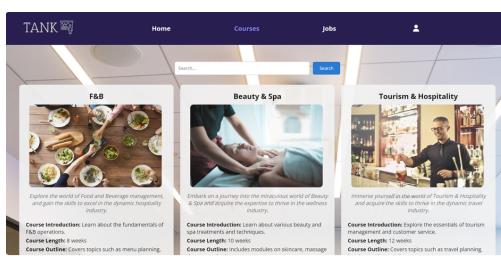
Home page



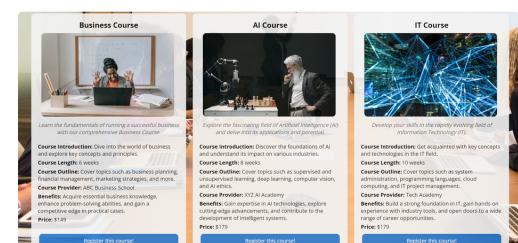
Register



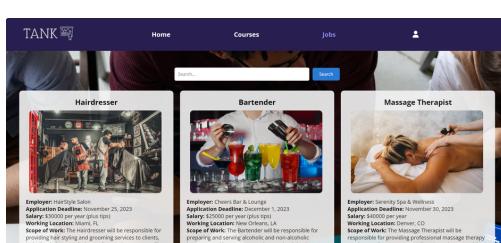
Login



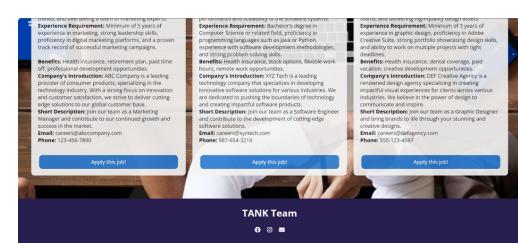
Courses page



Courses page



Jobs page



Jobs page

2. Database: all members

By utilizing MySQL as the RDBMS and PHPMyAdmin as the administration tool, we have created a well-structured and efficient database to support the software system. The database design process involved careful analysis, normalization, schema design, and indexing to optimize the performance (all mentioned and explained in previous pages).

Here we will only show some images of using PHPMyAdmin to manage the database:

The screenshot shows the PHPMyAdmin interface for the 'user_information' table. The table has columns: user_id, full_name, date_of_birth, gender, phone_number, email, and address. The data includes 10 rows of user information such as Juan Russo, Kevin Nokia, Logan Harrington, etc., with their respective details like birth dates, genders, phone numbers, emails, and addresses.

user_id	full_name	date_of_birth	gender	phone_number	email	address
2	Juan Russo	27/08/1977	Female	(404)970-830188390	marshamy@example.org	4444 Juniper Avenue, Valleyville, Greece
3	Kevin Nokia	09/12/2001	Female	+1-529-741-7365x275	brandohunter@example.org	3939 Sequoia Lane, Lakeshore, Switzerland*
4	Logan Harrington	08/11/1977	Male	662-839-3034x33499	ronald48@example.net	2222 Birch Avenue, Riverside, Vietnam
5	Matthew Barnes	22/12/1984	Male	916-567-4408	sonyachaney@example.com	456 Elm Avenue, Suburbia, Canada
6	Richard Newton	07/06/1980	Male	+1-267-285-5331	bill29@example.com	808 Spruce Lane, Hilltop, Mexico
7	Stacey Peru	10/06/1990	Male	590-499-0651x70733	johncjohnson@example.net	101 Pine Road, Hamlet City, Australia
8	Harold Middleton	01/09/1998	Female	725-625-0104	jasonpeterson@example.com	4848 Redwood Street, Lakeside, Romania
9	Rebecca Hansen	08/02/2002	Female	725-772-0991	kathyholtzman@example.net	1515 Maple Avenue, Highlands, South Korea
10	Kathleen Smith	03/07/2000	Female	321-503-6238x904	nrvnvlktu@example.com	1717 Willow Lane, Riverfront, France

A table showcasing user information

The screenshot shows the PHPMyAdmin interface for the 'job_description' table. The table has columns: job_id, job_title, recruiter, application_deadline, salary, working_location, category, and applicant_count. The data includes 10 rows of job descriptions such as HairStyle Salon, Cheers Bar & Lounge, Serenity Spa & Wellness, etc., with their respective details like recruiter names, application deadlines, salaries, locations, categories, and applicant counts.

job_id	job_title	recruiter	application_deadline	salary	working_location	category	applicant_count
1	Hairdresser	HairStyle Salon	November 25, 2023	30000.00	Silicon Valley, California	Education	40
2	Bartender	Cheers Bar & Lounge	December 1, 2023	25000.00	Silicon Valley, California	Education	40
3	Massage Therapist	Serenity Spa & Wellness	November 30, 2023	40000.00	Silicon Valley, California	Education	40
4	Marketing Manager	ABC Company	November 30, 2023	60000.00	Silicon Valley, California	Education	40
5	Software Engineer	XYZ Tech	December 15, 2023	80000.00	Silicon Valley, California	Education	40
6	Graphic Designer	DEF Creative Agency	December 10, 2023	50000.00	Silicon Valley, California	Education	40
7	Software Engineer	Microsoft	August 11, 2023	110000.00	Silicon Valley, California	IT	35
8	Marketing Manager	VMware	January 15, 2023	150000.00	Boston, Massachusetts	IT	22
9	Mechanical Engineer	IBM	January 15, 2023	75000.00	Silicon Valley, California	Business	31

A table showcasing job description

3. Data creation using Python: @Bui Thai Anh (Swinburne HN)

- Firstly, we used the Faker library and random library in the Python ecosystem to create much more dummy data we wanted.
- Secondly, we created some variables as tuples to store many random relevant data of the variables' names.
- Lastly, we used a FOR loop to create much more data as we expected.

```

from faker import Faker

fake = Faker()

usaUni = "University of Oxford", "University of Cambridge", "ETH Zurich - Swiss Federal Institute of Technology", "University College London (UCL)", "Imperial College London", "University of Edinburgh", "University of Manchester", "University of Munich", "Heidelberg University", "University of Amsterdam", "Sorbonne University", "Ecole Polytechnique", "University of Copenhagen", "University of Helsinki", "KU Leuven", "University of Zurich", "Stockholm University", "Ludwig Maximilian University of Munich", "University of Milan", "University of Oslo", "University of Geneva", "University of Bristol", "King's College London", "University of Lausanne", "University of Leipzig", "University of Bonn", "University of Würzburg", "University of Bologna", "University of Maastricht", "University of Groningen", "University of Antwerp", "University of Southampton", "University of Leipzig", "University of Toulouse", "University of Rennes 1", "University of Lisbon", "University of Porto", "University of Barcelona", "University of Seville", "University of Madrid", "University of Valencia", "Universitat de València", "Harvard University", "Stanford University", "Massachusetts Institute of Technology (MIT)", "California Institute of Technology (Caltech)", "Princeton University", "University of California, San Francisco (UCSF)", "University of Illinois Urbana-Champaign", "Northwestern University", "University of California, San Diego (UCSD)", "Rice University", "Vanderbilt University", "University of Colorado Boulder", "University of Arizona", "University of Rochester", "University of California, Los Angeles (UCLA)", "University of Miami", "University of Virginia Tech", "North Carolina State University", "Texas A&M University", "University of Delaware", "University of Connecticut", "Colorado State University", "University of Vermont", "Utah State University", "Auburn University", "Kansas State University", "University of Maine", "West Virginia University", "University of Mississippi", "University of Hawaii at Manoa", "University of Rhode Island", "Louisiana State University", "Mississippi State University", "University of Mississippi State University"

degreeUni = "Computer Science", "Mechanical Engineering", "Psychology", "Biochemistry", "Political Science", "Economics", "Environmental Science", "Graphic Design", "International Relations", "Neuroscience", "Journalism", "Fashion Design", "Marine Biology", "Civil Engineering", "Sociology", "Culinary Arts", "Linguistics", "Philosophy", "Urban Planning", "Public Health", "Aerospace Engineering", "Film Studies", "Data Science", "Biomedical Engineering", "Anthropology", "Digital Marketing", "Game Design", "Astrophysics", "Public Relations", "Healthcare Administration"

#print("SET @user_id = LAST_INSERT_ID();")
userID = 1
print("INSERT INTO educational_backgrounds (user_id, degree, institution)\n VALUES")
for i in range(600000):

    degree = fake.random_element(elements=(degreeUni))
    institution = fake.random_element(elements=(usaUni))

    print(f"({userID}), '{degree}', '{institution}'")
    userID += 1

    if i == 599999:
        print(f"({userID}), '{degree}', '{institution}'")
        break

```

Eg: This is a Python file for creating 600,000 rows of dummy data for the 'education_backgrounds' table in our database

However, we sometimes have difficulty in creating random data due to duplicate information or there is not much unique data. This is because of having created so much pre-trained data in variables. Therefore, we decided to add more libraries such as random so that it could help us to prevent from having duplicate data

```

courses.py X
D:\Py\coursespy>_
1  import random
2  from faker import Faker
3
4  fake = Faker()
5
6  #course name
7  courseName = ['Computer Science', 'Data Science', 'Web Development', 'Cybersecurity',
8  'Digital Marketing', 'Business Administration', 'Finance', 'Marketing', 'Project Management', 'Public Relations', 'Film Production', 'Graphic Design', 'Civil Engineering',
9  'Mechanical Engineering', 'Electrical Engineering', 'Chemical Engineering',
10 'Health Sciences', 'Nursing', 'Pharmacy', 'Public Health', 'Medicine', 'Physical Therapy', 'Education', 'Early Childhood Education', 'Curriculum and Instruction',
11 'Nutrition and Dietetics', 'Fitness and Wellness', 'Health Informatics', 'Teaching English as a Second Language (TESOL)', 'Linguistics and Language Education']
12
13 #course length
14 courseLength = ['10 weeks', '11 weeks', '12 weeks', '13 weeks', '14 weeks', '15 weeks', '16 weeks', '17 weeks', '18 weeks', '19 weeks', '20 weeks', '21 weeks',
15 '22 weeks', '23 weeks', '24 weeks', '25 weeks', '26 weeks', '27 weeks', '28 weeks', '29 weeks', '30 weeks', '31 weeks', '32 weeks', '33 weeks', '34 weeks', '35
16 '36 weeks', '37 weeks', '38 weeks', '39 weeks', '40 weeks']
17 #courseprice
18 coursePrice = ['152', '91', '267', '122', '75', '198', '105', '284', '68', '160', '234', '54', '117', '88', '203', '99', '77', '116', '215', '59', '270',
19 '176', '83', '124', '189', '72', '150', '195', '106', '63', '78', '282', '131', '59', '101', '271', '219', '113', '65', '88', '157', '285', '209', '79',
20 '273', '97', '252', '182', '54', '120', '66', '194', '282', '84']
21
22 #category
23 major = 'IT', 'Business', 'Media', 'Health', 'Education', 'Engineering'
24
25 #enrolment count
26 count = '5', '10', '15', '20', '25', '30', '35', '40', '45', '50'
27
28 print("INSERT INTO courses (course_name, course_length, course_price, category, enrolment_count)\n VALUES")
29 for i in range(600000):
30
31     course_name = fake.random.choice(courseName)
32     course_length = fake.random.choice(courseLength)
33     course_price = fake.random.choice(coursePrice)
34     category = fake.random_element(elements=(major))
35     enrolment_count = fake.random_element(elements=(count))
36
37     print(f"({course_name}), {course_length}, {course_price}, {category}, {enrolment_count},")
38

```

Eg: We decided to use a "random" library so we also had to modify code syntax to create much more unique dummy data for 'course_information' table in the database.

4. Back-end: @Bui Thai Anh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)

a) Create a connection to the database

```
connect.php X
connect.php > ...
1  <?php
2
3  $host_name = "localhost";
4  $username = "root";
5  $password = "";
6  $dbname = "tank";
7  $connect = mysqli_connect($host_name, $username, $password, $dbname);
8
9  // Connect to the database.
10 $connect = mysqli_connect($host_name, $username, $password, $dbname);
11
12 // Check if the connection was successful.
13 if (!$connect) {
14     die("Could not connect to the database: " . mysqli_connect_error());
15 }
16
17 // Connection was successful---->close it.
18 mysqli_close($connect);
19
20
21 ?>
```

connect.php

The `$host_name` variable is set to `localhost`, which means that the MySQL server is running on the same computer as the PHP script. The `$username` variable is set to `root`, which is the default username for the MySQL database. The `$password` variable is set to an empty string, which means that no password is required to connect to the database. The `$dbname` variable is set to `tank`, which is the name of the MySQL database to connect to.

The next line of code, `$connect = mysqli_connect($host_name, $username, $password, $dbname);`, connects to the database. The function takes the four database connection parameters as input and returns a connection object if the connection is successful. If the connection is not successful, the function returns `FALSE`.

The next line of code, `if (!$connect) {`, checks if the connection was successful. If the connection is not successful, the code dies with an error message.

The next line of code, `mysqli_close($connect);`, closes the database connection. This is important to do to avoid resource leaks.

This code is typically used at the beginning of a PHP script that needs to access the MySQL database. Once the connection is established, the script can execute SQL queries against the database.

b) Storing, sending, retrieving, and displaying information about course registration among users

```

processCourse.php
processCourse.php > ...
10 if ($_SERVER["REQUEST_METHOD"] == "POST") {
11     // Retrieve course data from the submitted form
12     $courseName = $_POST["course_name"];
13     $courseLength = $_POST["course_length"];
14     $coursePrice = $_POST["course_price"];
15     $category = $_POST["category"];
16     $enrolmentCount = $_POST["enrolment_count"];
17     $outlineText = $_POST["outline_text"];
18     $outcomeText = $_POST["outcome_text"];
19
20     // Insert data into the courses table
21     $sql = "INSERT INTO courses (course_name, course_length, course_price, category, enrolment_count)
VALUES ('$courseName', '$courseLength', '$coursePrice', '$category', '$enrolmentCount')";
22
23     if ($connect->query($sql) === TRUE) {
24         // Data inserted into courses table successfully
25
26         $course_id = $connect->insert_id; // Get the last inserted course_id
27
28         // Now, insert data into the course_outline table
29         $sqlOutline = "INSERT INTO course_outlines (course_id, outline_text) VALUES ('$course_id', '$outlineText')";
30         if ($connect->query($sqlOutline) === TRUE) {
31             // Data inserted into course_outline table successfully
32
33             // Finally, insert data into the course_outcomes table
34             $sqlOutcome = "INSERT INTO course_outcomes (course_id, outcome_text) VALUES ('$course_id', '$outcomeText')";
35             if ($connect->query($sqlOutcome) === TRUE) {
36                 echo "Success";
37             } else {
38                 echo "Error: " . $sqlOutcome . "<br>" . $connect->error;
39             }
40         } else {
41             echo "Error: " . $sqlOutline . "<br>" . $connect->error;
42         }
43     } else {
44         echo "Error: " . $sql . "<br>" . $connect->error;
45     }
46     header("Location: ./course.php");
47     exit();
48

```

processCourse.php

The first part of the code checks if the form has been submitted using the `$_SERVER["REQUEST_METHOD"] == "POST"` condition. If the form has been submitted, the code retrieves the course data from the submitted form using the `$_POST` variable.

The next part of the code inserts the course data into the `courses` table of the MySQL database using the `INSERT INTO` statement. The `$sql` variable contains the SQL query to insert the course data.

The `if ($connect->query($sql) === TRUE)` condition checks if the SQL query was executed successfully. If the query was executed successfully, the code gets the last inserted course ID using the `$connect->insert_id` property. The code then uses the course ID to insert the course outline and course outcomes into the `course_outlines` and `course_outcomes` tables, respectively.

The last part of the code redirects the user to the `/course.php` page.

Here is a step-by-step explanation of the code:

1. Check if the form has been submitted.
2. Retrieve the course data from the submitted form.
3. Insert the course data into the `courses` table.
4. Get the last inserted course ID.
5. Insert the course outline and course outcomes into the `course_outlines` and `course_outcomes` tables, respectively.
6. Redirect the user to the `/course.php` page.

This code is typically used in a PHP script that handles form submissions. When the user submits the form, the code inserts the form data into the database and then redirects the user to another page.

```

<div class="course-item">
    <h3>F&B</h3>
    
    <p class="course-description">Explore the world of Food and Beverage management, and gain the skills to excel in the dynamic hospitalit
    <ul class="course-details">
        <li><strong>Course Introduction:</strong> Learn about the fundamentals of F&B operations.</li>
        <li><strong>Course Length:</strong> 8 weeks</li>
        <li><strong>Course Outline:</strong> Covers topics such as menu planning, customer service, and restaurant management.</li>
        <li><strong>Course Provider:</strong> ABC Academy</li>
        <li><strong>Benefits:</strong> Develop practical skills and knowledge to succeed in F&B careers.</li>
        <li><strong>Price:</strong> $199</li>
    </ul>
    <a href="#" class="register-button">Register this course!</a>
    <form action="processCourse.php" method="post">
        <!-- Add hidden input fields to store course information -->
        <input type="hidden" name="course_name" value="F&B">
        <input type="hidden" name="course_length" value="8 weeks">
        <input type="hidden" name="course_price" value="199">
        <input type="hidden" name="category" value="Business">
        <input type="hidden" name="enrolment_count" value="25">
        <!-- Add other hidden input fields from course outline and course outcome table -->
        <input type="hidden" name="outline_text" value="Covers topics such as menu planning, customer service, and restaurant management.">
        <input type="hidden" name="outcome_text" value="Develop practical skills and knowledge to succeed in F&B careers.">
        <button type="submit" class="register-button">Register this course!</button>
    </form>
</div>

```

course.php

After that, there is a form tag in the image because the HTML code in the image is used to create a registration form for a course. The form tag tells the browser that the enclosed code is a form, and that the data submitted by the form should be sent to the server.

There is a form tag in the image because the HTML code in the image is used to create a registration form for a course. The form tag tells the browser that the enclosed code is a form, and that the data submitted by the form should be sent to the server.

The form tag also has a few attributes, including the `action` attribute and the `method` attribute. The `action` attribute specifies the URL of the server script that will process the form data. The `method` attribute specifies the HTTP method that will be used to submit the form data. In this case, the `action` attribute is set to `processCourse.php`, which means that the form data will be sent to the `processCourse.php` script on the server. The `method` attribute is set to `post`, which means that the form data will be submitted using the HTTP POST method.

The form tag contains a number of input fields, including a hidden input field, a text input field, and a submit button. The hidden input field is used to store the name of the course that the user is registering for. The text input field is used to collect the user's name. The submit button is used to submit the form data to the server.

When the user clicks the submit button, the browser will send the form data to the `processCourse.php` script on the server. The `processCourse.php` script will then process the form data and register the user for the course.

```

<?php
include_once("processCourse.php");

$sql2 = "SELECT * FROM courses
LEFT JOIN course_outlines ON courses.course_id = course_outlines.course_id
LEFT JOIN course_outcomes ON courses.course_id = course_outcomes.course_id";

$result2 = mysqli_query($connect, $sql2);
?>

```

courseDisplay.php

Lastly, in terms of displaying course information of users on a webpage called "courseDisplay", the first thing to do is to create a SQL query to retrieve data.

The next line of code, `include_once("processCourse.php")`, includes the `processCourse.php` file into the current script. This file may contain functions or classes that are used to process the course data.

The next line of code, `$sql2 = "SELECT * FROM courses LEFT JOIN course_outlines ON courses.course_id = course_outlines.course_id LEFT JOIN course_outcomes ON courses.course_id = course_outcomes.course_id";`, defines the SQL query that will be used to retrieve the course data.

```

<!-- Store data of courses that applicants buy-->
<?php while ($row_course = mysqli_fetch_assoc($result2)) : ?>
<div class="profile-container">
<div class="profile-image">
|   
</div>
<div class="profile-details">
<?php
echo "<h2>Course ".$row_course["course_id"]."</h2>";
?>
<?php
echo "<p><strong>Course Name:</strong> ".$row_course["course_name"]."</p>";
echo "<p><strong>Course Length:</strong> ".$row_course["course_length"]."</p>";
echo "<p><strong>Course Price:</strong> ".$row_course["course_price"]."</p>";
echo "<p><strong>Category:</strong> ".$row_course["category"]."</p>";
echo "<p><strong>Enrolment Count of course:</strong> ".$row_course["enrolment_count"]." participants </p>";
?>
</div>

<div class="section-heading"></div>

<div class="profile-details">
<h2>Course Outline</h2>
<?php
echo "<p><strong>About this Course:</strong></p>";
echo "<p>".$row_course["outline_text"]."</p>";
?>
</div>

<div class="section-heading"></div>

<div class="profile-details">
<h2>Course Outcome</h2>
<?php
echo "<p><strong>After this course, you can:</strong></p>";
echo "<p>".$row_course["outcome_text"]."</p>";
?>
</div>

```

courseDisplay.php

Finally, let's display the information of courses purchased by users.

This line starts a `while` loop that fetches each row from the result set (`$result2`) and assigns it to the associative array `$row_course`. The loop continues as long as there are more rows to fetch. In each part, we used `$row_course["(name of each column in the Courses table in the database)"]` to display the data on a webpage so that a user can observe his/her registered courses.

The screenshot shows a web browser window with the URL `localhost/cos20031/courseDisplay.php`. The page displays two sections of course information for different users.

Course 1:

- Course Name:** Beauty & Spa
- Course Length:** 10 weeks
- Course Price:** \$249.00
- Category:** Business
- Enrolment Count of course:** 26 participants

Course Outline:

About this Course:
Includes modules on skincare, massage therapy, and salon management.

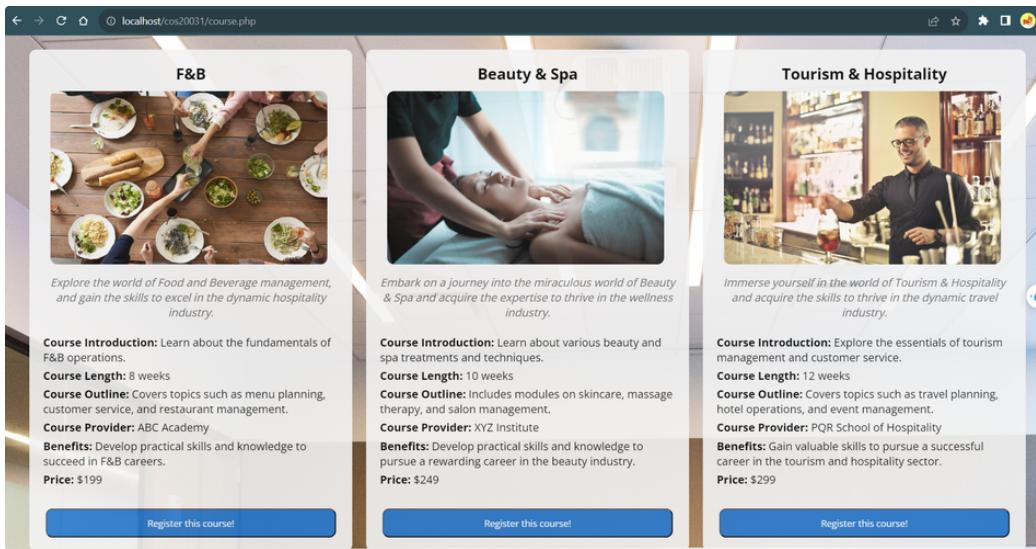
Course Outcome:

After this course, you can:
Develop practical skills and knowledge to pursue a rewarding career in the beauty industry.

Course 2:

- Course Name:** Business Course
- Course Length:** 6 weeks
- Course Price:** \$149.00
- Category:** Business
- Enrolment Count of course:** 28 participants

Displaying some courses a user purchased



Some information about some courses

c) Storing, sending, retrieving, and displaying information about job applications among users

```


|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div class="job-item"> <h3>Hairdresser</h3>  <p><strong>Employer:</strong> HairStyle Salon</p> <p><strong>Application Deadline:</strong> November 25, 2023</p> <p><strong>Salary:</strong> \$30000 per year (plus tips)</p> <p><strong>Working Location:</strong> Miami, FL</p> <p><strong>Scope of Work:</strong> The Hairdresser will be responsible for providing hair styling and grooming services to clients, including</p> <p><strong>Experience Requirement:</strong> Minimum of 2 years of experience as a professional hairdresser, knowledge of current hair trends</p> <p><strong>Benefits:</strong> Flexible work hours, discounted salon services, opportunities for professional development and training.</p> <p><strong>Company's Introduction:</strong> HairStyle Salon is a reputable salon that offers various hair care services to clients. We have a</p> <p><strong>Short Description:</strong> Join our team as a Hairdresser and showcase your creativity and skills while making clients look and f</p> <p><strong>Email:</strong> careers@hairstylesalon.com</p> <p><strong>Phone:</strong> 555-123-4567</p> <a class="apply-button" href="#">Apply this job!</a> </div> | <div class="job-item"> <p><strong>Course Introduction:</strong> Learn about the fundamentals of F&amp;B operations.</p> <p><strong>Course Length:</strong> 8 weeks</p> <p><strong>Course Outline:</strong> Covers topics such as menu planning, customer service, and restaurant management.</p> <p><strong>Course Provider:</strong> ABC Academy</p> <p><strong>Benefits:</strong> Develop practical skills and knowledge to succeed in F&amp;B careers.</p> <p><strong>Price:</strong> \$199</p> </div> | <div class="job-item"> <p><strong>Course Introduction:</strong> Learn about various beauty and spa treatments and techniques.</p> <p><strong>Course Length:</strong> 10 weeks</p> <p><strong>Course Outline:</strong> Includes modules on skincare, massage therapy, and salon management.</p> <p><strong>Course Provider:</strong> XYZ Institute</p> <p><strong>Benefits:</strong> Develop practical skills and knowledge to pursue a rewarding career in the beauty industry.</p> <p><strong>Price:</strong> \$249</p> </div> | <div class="job-item"> <p><strong>Course Introduction:</strong> Explore the essentials of tourism management and customer service.</p> <p><strong>Course Length:</strong> 12 weeks</p> <p><strong>Course Outline:</strong> Covers topics such as travel planning, hotel operations, and event management.</p> <p><strong>Course Provider:</strong> PQR School of Hospitality</p> <p><strong>Benefits:</strong> Gain valuable skills to pursue a successful career in the tourism and hospitality sector.</p> <p><strong>Price:</strong> \$299</p> </div> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|


```

job.php

First of all, there is a form tag in the image because the HTML code in the image is used to create a registration form for a job. The form tag tells the browser that the enclosed code is a form, and that the data submitted by the form should be sent to the server.

The form tag also has a few attributes, including the `action` attribute and the `method` attribute. The `action` attribute specifies the URL of the server script that will process the form data. The `method` attribute specifies the HTTP method that will be used to submit the form data. In this case, the `action` attribute is set to `processJob.php`, which means that the form data will be sent to the `processJob.php` script on the server. The `method` attribute is set to `post`, which means that the form data will be submitted using the HTTP POST method.

The form tag contains a number of input fields, including a hidden input field, a text input field, and a submit button. The hidden input field is used to store the name of the course that the user is registering for. The text input field is used to collect the user's name. The submit button is used to submit the form data to the server.

When the user clicks the submit button, the browser will send the form data to the `processJob.php` script on the server. The `processJob.php` script will then process the form data and register the user for the course.

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $jobTitle = $_POST["job_title"];
    $recruiter = $_POST["recruiter"];
    $applicationDeadline = $_POST["application_deadline"];
    $salary = $_POST["salary"];
    $workingLocation = $_POST["working_location"];
    $category = $_POST["category"];
    $applicantCount = $_POST["applicant_count"];
    $requirement = $_POST["requirement"];
    $benefit = $_POST["benefit"];

    // Insert data into the job_description table
    $sql = "INSERT INTO job_description (job_title, Recruiter, application_deadline, salary, working_location, category, applicant_count)
VALUES ('$jobTitle', '$recruiter', '$applicationDeadline', '$salary', '$workingLocation', '$category', '$applicantCount')";

    if ($connect->query($sql) === TRUE) {
        $job_id = $connect->insert_id; // Get the last inserted job_id

        // Now, insert data into the job_requirements table
        $sqlRequirement = "INSERT INTO job_requirements (job_id, requirement) VALUES ('$job_id', '$requirement')";
        if ($connect->query($sqlRequirement) === TRUE) {
            // Finally, insert data into the job_benefits table
            $sqlBenefit = "INSERT INTO job_benefits (job_id, benefit) VALUES ('$job_id', '$benefit')";
            if ($connect->query($sqlBenefit) === TRUE) {
                echo "Success";
            } else {
                echo "Error: " . $sqlBenefit . "<br>" . $connect->error;
            }
        } else {
            echo "Error: " . $sqlRequirement . "<br>" . $connect->error;
        }
    } else {
        echo "Error: " . $sql . "<br>" . $connect->error;
    }
    header("Location: ./job.php");
    exit();
}

```

processJob.php

The code line if (\$_SERVER["REQUEST_METHOD"] == "POST") { checks if the form is submitted using the POST method. Besides, \$variable = \$_POST["(name of columns in the job_description table)"], this syntax is used to retrieve various form input values using `$_POST` and assigns them to corresponding variables.

We used a SQL query to insert data into the table in the database \$sql = "INSERT INTO job_description (job_title, Recruiter, application_deadline, salary, working_location, category, applicant_count)

VALUES ('\$jobTitle', '\$recruiter', '\$applicationDeadline', '\$salary', '\$workingLocation', '\$category', '\$applicantCount');

\$job_id = \$connect->insert_id;> This syntax is used to get the latest job_id

if (\$connect->query(\$sqlRequirement) === TRUE) { ----> This syntax is used to check if the job_requirements insertion was successful. If the requirements insertion was successful, it proceeds to insert job benefits. We also use a SQL query to insert job benefits into the job_benefits table \$sqlBenefit = "INSERT INTO job_benefits (job_id, benefit) VALUES ('\$job_id', '\$benefit')";

```

jobDisplay.php x
jobDisplay.php > HTML > body > header > div.nhan1
1  <?php
2  include_once("processJob.php");
3
4  $sql2 = "SELECT * FROM job_description
5  LEFT JOIN job_requirements ON job_description.job_id = job_requirements.job_id
6  LEFT JOIN job_benefits ON job_description.job_id = job_benefits.job_id";
7
8  $result2 = mysqli_query($connect, $sql2);
9  ?>
10 <!DOCTYPE html>

```

jobDisplay.php

The line of code, \$sql2 = "SELECT * FROM job_description LEFT JOIN job_requirements ON job_description.job_id = job_requirements.job_id LEFT JOIN job_benefits ON job_description.job_id = job_benefits.job_id";, defines the SQL query that will be used to retrieve the job data.

```

jobDisplay.php X
jobDisplay.php ?...
40      <!--Store data of courses that applicants buy-->
41      <?php while ($row_job = mysqli_fetch_assoc($result2)) : ?>
42      <div class="profile-container">
43          <div class="profile-image">
44              | 
45          </div>
46          <div class="profile-detail">
47              <?php
48                  echo "h2>Job ".$row_job["job_id"]."</h2>";
49              ?>
50              <?php
51                  echo "<p><strong>Job Title:</strong> ".$row_job["job_title"]."</p>";
52                  echo "<p><strong>Employer:</strong> ".$row_job["recruiter"]."</p>";
53                  echo "<p><strong>Application Deadline:</strong> ".$row_job["application_deadline"]."</p>";
54                  echo "<p><strong>Salary:</strong> ".$row_job["salary"]." per year.</p>";
55                  echo "<p><strong>Working location:</strong> ".$row_job["working_location"]."</p>";
56                  echo "<p><strong>Category of this job:</strong> ".$row_job["category"]."</p>";
57                  echo "<p><strong>Quantity of applicants:</strong> ".$row_job["applicant_count"]." participants </p>";
58              ?>
59          </div>
60      </div>
61      <div class="section-heading"></div>
62
63      <div class="profile-details">
64          <h2>Job Requirement</h2>
65          <?php
66              echo "<p><strong>Before applying, you must:</strong></p>";
67              echo "<p>".$row_job["requirement"]."</p>";
68          ?>
69      </div>
70
71      <div class="section-heading"></div>
72
73      <div class="profile-details">
74          <h2>Job Benefits</h2>
75          <?php
76              echo "<p><strong>Becoming a member of our company, you have:</strong></p>";
77              echo "<p>".$row_job["benefit"]."</p>";
78          ?>
79      </div>

```

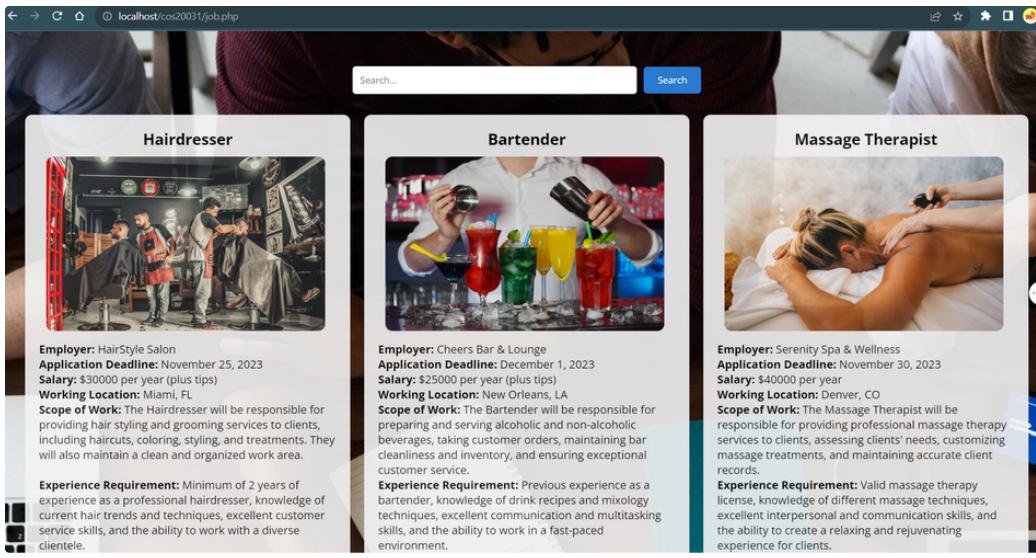
jobDisplay.php

Finally, let's display the information of jobs applied by users.

This line starts a `while` loop that fetches each row from the result set (`$result2`) and assigns it to the associative array `$row_job`. The loop continues as long as there are more rows to fetch. In each part, we used `$row_job["(name of each column in the job_description table in the database)"]` to display the data on a webpage so that a user can observe his/her applied occupations.



The screenshot shows a web page titled "localhost/cos20031/jobDisplay.php". The page displays two job listings, each with a circular profile picture at the top. Job 1 is for a "Massage Therapist" at "Serenity Spa & Wellness" with a deadline of "November 30, 2023". Job 2 is for a "Hairdresser". Below each job listing is a "Job Requirement" section with a note about valid massage therapy license requirements, and a "Job Benefits" section mentioning health insurance and training. A footer at the bottom of the page says "Display some jobs that a user applied".



Some information about jobs available

d) Storing, sending, retrieving, and displaying personal information of users

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $fullname = $_POST["fullname"];
    $email = $_POST["email"];
    $dateofbirth = $_POST["dateofbirth"];
    $phonenumber = $_POST["phonenumber"];
    $gender = $_POST["gender"];
    $address = $_POST["address"];
    $degree = $_POST["degree"];
    $institution = $_POST["institution"];
    $jobtitle = $_POST["jobtitle"];
    $employer = $_POST["employer"];

    // Insert data into the user_information table
    $sql = "INSERT INTO user_information (full_name, date_of_birth, gender, phone_number, email, address)
VALUES ('$fullname', '$dateofbirth', '$gender', '$phonenumber', '$email', '$address')";

    if ($connect->query($sql) === TRUE) {
        // Data inserted into user_information table successfully
        $user_id = $connect->insert_id; // Get the last inserted user_id

        // Now, insert data into the user_login table
        $username = $_POST["username"];
        $password = $_POST["password"];

        $sql = "INSERT INTO user_login (username, password, user_id) VALUES ('$username', '$password', '$user_id')";

        if ($connect->query($sql) === TRUE) {
            // Data inserted into user_login table successfully

            // Now, insert data into the educational_backgrounds table
            $sql = "INSERT INTO educational_backgrounds (user_id, degree, institution) VALUES ('$user_id', '$degree', '$institution')";
            if ($connect->query($sql) === TRUE) {
                // Data inserted into educational_backgrounds table successfully

                // Finally, insert data into the working_experiences table
                $sql = "INSERT INTO working_experiences (user_id, job_title, employer) VALUES ('$user_id', '$jobtitle', '$employer')";
                if ($connect->query($sql) === TRUE) {
                    echo "Success";
                }
            }
        }
    }
}

```

register.php

In the user registration process, our initial step involves the collection of personal information through a dedicated form when users sign up on the website. Upon successful registration, users will be seamlessly directed to the homepage, ensuring a smooth and user-friendly experience. Subsequently, the gathered user data, including personal details, will be securely stored in a database.

The method we used in this situation is as same as the way we implemented on developing backend system for course and job management.

We developed a backend system, which will empower the website to dynamically display user information on a designated webpage called "Profile," providing users with a centralized and accessible view of their registered details. This comprehensive approach ensures both data security and a streamlined user interface, contributing to a positive and efficient user interaction with the website.

```

<?php
include("connect.php");
include("processCourse.php");

$sql = "SELECT * FROM user_information , working_experiences,
           | educational_backgrounds;";

$sql2 = "SELECT * FROM courses
        LEFT JOIN course_outlines ON courses.course_id = course_outlines.course_id
        LEFT JOIN course_outcomes ON courses.course_id = course_outcomes.course_id;"

$result = mysqli_query($connect, $sql);
$result2 = mysqli_query($connect, $sql2);

//Display data about user information
if ($result){
    while ($row = mysqli_fetch_assoc($result)){
        //user information table
        $id = $row["user_id"];
        $fullname = $row["full_name"];
        $dateofbirth = $row["date_of_birth"];
        $gender = $row["gender"];
        $phonenumber = $row["phone_number"];
        $email = $row["email"];
        $address = $row["address"];

        //educational background table
        $degree = $row["degree"];
        $institution = $row["institution"];

        //working experience table
        $jobtitle = $row["job_title"];
        $employer = $row["employer"];
    }
}

//Display data about courses registration
//    -> while loop in coursesdisplay page

```

profile.php

This part of the source code checks if the database query result (`$result`) is valid (not false) and then enters a while loop. The `mysqli_fetch_assoc` function is used to fetch the next row from the result set as an associative array (`$row`).

```

<!--Create a profile page to store data of an applicant-->
<div class="profile-container">
    <div class="profile-image">
        |   
    </div>
    <div class="profile-details">
        <h2>User Information</h2>
        <?php
        echo "<p><strong>Full Name:</strong> ".$fullname."</p>";
        echo "<p><strong>Date of birth:</strong> ".$dateofbirth."</p>";
        echo "<p><strong>Gender:</strong> ".$gender."</p>";
        echo "<p><strong>Phone Number:</strong> ".$phonenumber."</p>";
        echo "<p><strong>Email:</strong> ".$email."</p>";
        echo "<p><strong>Address:</strong> ".$address."</p>";
        ?>
    </div>

    <div class="section-heading"></div>

    <div class="profile-details">
        <h2>Education Background</h2>
        <?php
        echo "<p><strong>Degree:</strong> ".$degree."</p>";
        echo "<p><strong>Institution:</strong> ".$institution."</p>";
        ?>
    </div>

    <div class="section-heading"></div>

    <div class="profile-details">
        <h2>Working Experience</h2>
        <?php
        echo "<p><strong>Job title:</strong> ".$jobtitle."</p>";
        echo "<p><strong>Employer:</strong> ".$employer."</p>";
        ?>
    </div>
</div>

```

profile.php

Finally, we simply used \${variable} to display the data of user information after filling a register form at the beginning of visiting our website.

The screenshot shows a user profile page with the following sections:

- User Information**
 - Full Name: BinCr7
 - Date of birth: 2023-11-10
 - Gender: Male
 - Phone Number: 123
 - Email: bin@gmail.com
 - Address: Japan
- Education Background**
 - Degree: Acting
 - Institution: Havard University
- Working Experience**
 - Job title: Software Engineer
 - Employer: Steve Job

personal information of a user

Team Retrospective

📋 Overview

Reflect back on what you and your team learned and what motivates the group to succeed by following the instructions for the [4Ls Retrospective Play](#).

Team	Team TANK
Team members	@Nguyen Duy Anh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN) @Bui Thai Anh (Swinburne HN) @Lai Gia Khanh (Swinburne HN)
Date	21 Nov 2023
Retrospective period	The entire semester

💡 4Ls retrospective

Milestones	Loved	Longed for	Loathed	Learned
Team Collaboration	Appreciated how well we communicated and shared ideas, which resulted in rapid problem-solving and a strong sense of team unity.	Longed for more regular knowledge sharing sessions and cross-functional collaboration to expand our collective expertise and ideas.	Found it challenging when some team members did not actively participate in conversations and decision-making processes, causing delays in progress.	Learned the essence of regular communication and active participation of all team members so as to foster a collaborative and productive work environment.
Project management using Confluence and Jira	Confluence and Jira simplified our project management and documentation procedures, making it simple to track progress and collaborate on assignments.	Wish we had a better understanding of these platforms earlier so that we could easily meet the project's requirements.	Complex configuration and customization: quite complicated and time-consuming for newcomers to comprehend.	Learned how to efficiently use Confluence and Jira's numerous features to organize tasks, track progress, and maintain a consolidated knowledge library.
Web Development	Loved the creativity and innovation we brought to the web development process, resulting in a visually appealing and user-friendly website.	Wish we had better skills in doing backend functions (such as writing an API for our website).	Unexpected technical issues, bugs and tight deadlines.	Front-end development, back-end development, managing database, dealing with server requests and responses.

Database Design	Quite pleased with how meticulously we planned and organized the database design, assuring quick data access.	Better management of complicated data, as well as enhanced analytics.	A large amount of dummy data needs to be created, which is quite time-consuming. However, this challenge provided us with an opportunity to practice and implement indexes efficiently.	Create tables, primary, foreign keys; create 1NF-3NF database and choose the optimal one; learned new SQL statements and the way to utilize indexes.
-----------------	---	---	---	--

⚡ Action plan

Action	Owner	Due date	Action items
Team Collaboration	@Nguyen Duc Nhan (Swinburne HN) @Bui Thai Anh (Swinburne HN) @Lai Gia Khanh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	26 Nov 2023	Confluence, Jira, Messenger, Google Meet
Database Design	@Nguyen Duc Nhan (Swinburne HN) @Bui Thai Anh (Swinburne HN) @Lai Gia Khanh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	12 Nov 2023	Physical Database Document on data creation and null values Query command (Use cases) Performance (Indexes)
Website Development (Front-end, Back-end)	@Nguyen Duc Nhan (Swinburne HN) @Bui Thai Anh (Swinburne HN) @Lai Gia Khanh (Swinburne HN) @Nguyen Duy Anh (Swinburne HN)	15 Nov 2023	HTML, CSS, PHP, Python
Code Debug	@Nguyen Duc Nhan (Swinburne HN) @Bui Thai Anh (Swinburne HN)	15 Nov 2023	HTML, CSS, PHP, Python
Slides	@Nguyen Duy Anh (Swinburne HN) @Lai Gia Khanh (Swinburne HN) @Nguyen Duc Nhan (Swinburne HN) @Bui Thai Anh (Swinburne HN)	25 Nov 2023	Google Slides

Check all slides	@Nguyen Duc Nhan (Swinburne HN)	25 Nov 2023	Google Slides
Edit video	@Bui Thai Anh (Swinburne HN)	26 Nov 2023	CapCut
Check all documents on Confluence and Jira	@Nguyen Duc Nhan (Swinburne HN)	26 Nov 2023	Confluence, Jira

Jira Timeline

		SEP	OCT	NOV					
Sprints		Get ac...	Project Proposal	Initial database l...	Progre...	Frontend	Enhanc...	Backend	Final run-up
> TANK-1 Project Proposal	DONE								
> TANK-9 Initial database	DONE								
> TANK-12 Progress Report	DONE								
> TANK-13 Frontend	DONE								
> TANK-51 Enhanced database	DONE								
> TANK-16 Backend	DONE								
> TANK-26 Process and Product Video	DONE								
> TANK-25 Final Report and Product Deliverable	DONE								

The completed Jira timeline

		SEP	OCT	NOV					
Sprints		Get ac...	Project Proposal	Initial database l...	Progre...	Frontend	Enhanc...	Backend	Final run-up
> TANK-1 Project Proposal	DONE								
TANK-2 Working Agreement	DONE	BUI THAI...							
TANK-4 Provision for teammate under...	DONE	NGUYEN...							
TANK-3 Project Plan	DONE	NGUYEN...							
TANK-4 Roles and Responsibilities	DONE	DUC NHA...							
TANK-5 Risk Assessment	DONE	DUC NHA...							
TANK-6 Persona	DONE	LAI GIA K...							
TANK-7 Empathy Map	DONE	LAI GIA K...							
TANK-8 Product requirement	DONE	NGUYEN...							
TANK-11 ER diagram	DONE	BUI THAI...							
TANK-49 Submit the Project Proposal	DONE	DUC NHA...							
> TANK-9 Initial database	DONE								
TANK-12 Updated ER diagram	DONE	BUI THAI...							
TANK-45 Create table statements	DONE	NGUYEN...							
> TANK-12 Progress Report	DONE								
TANK-20 Fixing Project Plan	DONE	NGUYEN...							
TANK-21 Redo Product Requirement	DONE	LAI GIA K...							
TANK-22 Team Health Monitor	DONE	DUC NHA...							
TANK-54 Submit the Progress Report	DONE	DUC NHA...							
> TANK-13 Frontend	DONE								
TANK-29 Header/footer	DONE	DUC NHA...							
TANK-14 User registration	DONE	DUC NHA...							
TANK-28 Login	DONE	DUC NHA...							
TANK-35 Home page	DONE	LAI GIA K...							
TANK-15 Course selection	DONE	LAI GIA K...							
TANK-49 Job selection	DONE	DUC NHA...							
> TANK-51 Enhanced database	DONE								
TANK-46 Insert dummy data	DONE	BUI THAI...							
TANK-53 Use cases and SQL statement...	DONE	NGUYEN...							
TANK-52 Performance (Indexes)	DONE	BUI THAI...							
> TANK-16 Backend	DONE								
TANK-30 User registration	DONE	NGUYEN...							
TANK-32 Register courses	DONE	BUI THAI...							
TANK-33 Apply jobs	DONE	BUI THAI...							
TANK-34 Profile update	DONE	BUI THAI...							
> TANK-26 Process and Product Video	DONE								
TANK-55 Introduction, project backgrou...	DONE	DUC NHA...							
TANK-56 Mission statement and objectiv...	DONE	NGUYEN...							
TANK-57 Team process	DONE	LAI GIA K...							
TANK-58 Logical/physical design	DONE	BUI THAI...							
TANK-59 Discussion of solutions to use c...	DONE	NGUYEN...							
TANK-60 Major-specific technology	DONE	DUC NHA...							
TANK-61 Summary of solution and lesso...	DONE	LAI GIA K...							
TANK-44 Edit videos	DONE	BUI THAI...							
> TANK-25 Final Report and Product Deliverable	DONE								
TANK-36 Review of ER diagram	DONE	NGUYEN...							
TANK-37 Physical database	DONE	LAI GIA K...							
TANK-38 Use cases and SQL statements	DONE	NGUYEN...							
TANK-62 Document on data creation an...	DONE	BUI THAI...							
TANK-39 Performance (Indexes)	DONE	BUI THAI...							
TANK-40 Major-specific work	DONE	DUC NHA...							
TANK-41 Team reflection	DONE	DUC NHA...							

The completed Jira timeline with all tasks