ASSESSMENT BRIEF

MODULE CODE	COM4003
MODULE TITLE	Introduction to Tech Stack
MODULE LEADER	Dr Yashar Baradaranshokouhi
ASSESSMENT TITLE	Project Artefact
WEIGHTING	60%

INSTRUCTIONS

The assessment is designed to evaluate:

- Your ability to implement a basic but working version of the application/prototype, containing only the core functional components, but covering all the layers of the tech stack.
- Your understanding of the different technologies involved in the development of the application.
- Your knowledge and the technical skills you have acquired in the module to select and implement a technology stack for a given application.

SCENARIO

As part of this assessment, you should develop a job board web application, and document the application requirements/specifications, and installation. The web application should connect prospective employers with students.

You must choose an appropriate tech stack, including database, server-side language and framework, and user interface frameworks. You should create a basic implementation that showcases the main features so that you can run a pilot test to evaluate the idea. From this initial version, the application needs to be easy to grow and scale.

A Minimum Viable Product (MVP) is provided in the next section. This should not be considered the minimum work that you should complete to pass the module. There are other criteria that you need to complete to pass the module. Please refer to the marking criteria for a comprehensive list of criteria and specifications for each criterion.

WHAT YOU NEED TO DO

Here are some steps to guide you through your project.

1. Identify and explore previous works consisting of various websites, online services, software and online portals that offer similar functionalities most relevant to the current project, and extract a list of features and functionalities to embed, integrate and implement in your project.
2. Research different technology stacks for the rest of the project, referring to content from earlier sessions, and select the back-end technologies.
3. Select and justify back-end technologies and frameworks
4. Implement the user interface (you can use mock data).
5. Develop a project plan where you describe the technologies you have chosen and how you will use them to complete the project. The plan should include a comprehensive list of tasks and a schedule.
6. For the user interface, you should submit a link to an active git repository containing your project code, as well as an archived version of the same repository. The zip file should contain all components of your work, as well as the documentation.
7. Your project repository should contain the required information to be able to reproduce your development environment and run your application. For example, your repo should include a README.md file with basic installation and usage instructions; and requirements.
8. Use the following GitHub Classroom repo for the purpose of the assessment. https://classroom.github.com/a/v5aGC2US

RECORDED VIDEO PRESENTATION

- Prepare and record a presentation of the user journey of the website and explain the key functionalities and features of your work

MVP (MINIMUM VIABLE PRODUCT):

The MVP should be fully functional in terms of the core features described. It should be free of critical bugs that prevent users from creating profiles, searching, or browsing. User data should be handled securely, especially authentication and personal information.

STUDENT PERSPECTIVE

- Students can register and create a profile with the following fields:
  - Full name
  - Contact information (email, phone number)
  - Upload, edit or delete a photograph (if uploading a new photo, add the option to delete the previous one)
  - Key skills (a list of tags)
  - Academic timeline (list of educational institutions, fields of study, years attended)
  - Work experience timeline (list of positions, companies, responsibilities, duration)
- Ability to update their profiles.
- Students should be able to search for available jobs.

EMPLOYER PERSPECTIVE

- Employers can register and create a profile with the following fields:
  - Company Name
  - Contact information (email, website address)
  - Photograph upload feature (add/delete/edit)
- Employers can search for student profiles using key skills or other relevant keywords.
- Employers can advertise new positions/opportunities and close old job adverts.

DATABASE DESIGN & IMPLEMENTATION

- A basic database implementation
  This will most likely involve the inclusion of .sql or .json files from which the database can be restored/recreated.
  - User/student profile
  - Employer profile
  - Job posting
  - Application
- Testing and documentation (refer to the ReadMe section)

COMMON FEATURES

- The application should be suitable for both desktop and mobile browsers.
- Secure login and registration process for both students and employers.
- A simple landing page that directs users to either log in/register or continue to the job board as a guest to browse profiles.

DOCUMENTATION (ReadMe)

Create a ReadMe file consisting of the following sections:

- Use Git for tracking changes and feedback from the module leader (Link) Include the link to the GitHub repo in the ReadMe file

- Discuss the use of cloud platforms such as Heroku or Azure for easy deployment and scaling.
- Project overview
  - Project goals and objectives
  - Scope of the project
- Installation instructions
  - A step-by-step guide that explains how to set up the development environment, how to install dependencies, and how to get the application running on a local machine or server.
  - for a user to be able to setup the database. This should include the instructions needed to implement the database. The documentation should cover the core functionality and demonstrate the integration with the middleware.
- The basic architecture of the application middleware. While you aren't required to implement all the functional components of the app, you should be able to move data all the way from the database to the interface for some core functions. This will be enabled by the middleware server-side code.
- Legal and ethical consideration
- Risk assessment
- Future considerations for scaling
- Project plan & reflection
  - Timeline of developing features/requirements
    - Front-end/user interface in semester 1
    - Back-end/server-side in semester 2
  - Milestones
  - Reflect on your project planning and implementation of the project.
- In-code documentation for key functions and components

Ensure that your project plan is detailed and realistic, accounting for the scope of the project and your academic calendar. Adjustments to the plan may be necessary as the project progresses, and such changes should be documented and communicated effectively.

SUBMISSION DETAILS

RELEASE DATE	16th of February 2024
SUBMISSION DATE	28th of March 2024 (noon 12:00)
DELIVERABLES	• The ReadMe(.md) should consist of a document with the timeline, milestones, scope and objectives, etc. as outlined in the MVP section above. • All student work (latest version) should be uploaded to the assessment portal on Moodle as a single compressed file (.zip file recommended) • All students must accept the allocated repository on GitHub Classroom for the assessment and push/sync all their work to the allocated repository before the deadline.
SUBMISSION DETAILS	Submit your assignment by uploading it to Moodle by midday on the submission date. This deadline will be automatically and strictly enforced. If your submission is late, your grade may be affected. If you have any issues submitting your work, you must email the assessment team and copy in the module leader before the assessment due time. Do not leave your submission until the last minute to avoid any penalties due to problems with the submission portal. Assessment team: assessment@leedstrinity.ac.uk Module Leader: y.b.shokouhi@leedstrinity.ac.uk We may ask for a demonstration of your work following the submission. If needed, this will be communicated to you individually via email. Please check your emails regularly.

Your feedback and mark for this assessment will be provided within 15 working days.

MARKING CRITERIA

Marks are awarded based on the following criteria. Within each part, aim to complete the work for each section before moving on to the next. The following banded marking scheme is used:

Exceptional 1st	100/95/92		*2:ii*	58/55/52
Outstanding 1st	88/85/82		*3rd*	48/45/42
1st	78/75/72		*Bare Fail*	38/35/32
2:i	68/65/62		*Fail*	25/20/10/0

To obtain a 3rd mark (40%)	o Ideas are adequately expressed, and the reader can follow the rationale and mechanical thinking of the developer, but it lacks critical discussion and depth. o The MVP is delivered with a justification for any missing features. o Subject-specific skills, including logical thinking and programming abilities, are satisfactory. A foundation level of technical skills and knowledge necessary for web application development is demonstrated in the documentation and submitted project artefact. o The code is executable with some debugging indicating a need for further refinement and testing. o The project structure and code organisation may need further improvement. The structure of the project may be untidy, and it makes scaling challenging. o The project documentation conveys a factual approach, showing broadly accurate knowledge within the context of the project. o Legal and ethical considerations have been discussed in a basic manner, demonstrating a general level of awareness of key areas. o There seem to be some links to real-world applications. o A generic level of risk assessment of the project has been carried out and the documentation is missing discussions around containing the risk factors. o Version control is utilised and project files are available on GitHub.
To obtain a 2:ii mark (50%), you must have (in addition to the above):	o Critical discussion around real-world applications, with strong links to a reliable source of information published research work. A minimum of 4 appropriate and relevant references must be cited. o Good engagement with an appropriate range of reading beyond essential texts. Statements are supported with references to reliable source of information, but referencing may show minor inaccuracies or inconsistencies. o A good level of subject-specific skills and practices have been demonstrated, but some minor flaws are evident. o The ReadMe file holds some instructions and guidelines on how to replicate the steps to create the project structure and execute the project. o The project meets the requirements specified in the MVP. Some parts of the code need further improvement, but the core part of the project is functional. o The code is reasonably well organized. There is little unused or irrelevant code, or this code has been moved out of the main project files. Variable and function names are generally meaningful and helpful for understanding. o Minimal database schema design and implementation of the database. Some effort to reduce the data redundancy, but not fully normalised.

	○ Risk factors discussed along with supporting evidence of practical strategies or techniques applied to mitigate or manage the risks.
To obtain a 2:i mark (60%), you must have (in addition to the above):	○ Very good level of knowledge of techniques and technology in the context of the project. ○ showing confidence and understanding of the basic underlying concepts and principles in software development. ○ Consistently accurate application of referencing. A minimum of 7 reliable and relevant refences to be cited. ○ Code has been well-commented and optimised leading to better use of resources. ○ The project structure is very well organized. No irrelevant or distracting code/files. Coding conventions and styles have been applied. The code is commented throughout, and it is easy to read and understand. ○ The documentation demonstrates a good level of understanding of the techniques and algorithms. ○ Excellent level of depth and insight with critical discussion around legal and ethical considerations. ○ The project exceeds the expectations in the MVP and delivers a high-quality semi-professional level product. ○ A visual presentation of the project timeline (Gantt Chart) with evidence of tracking progress on the milestones is provided in the project documentation. ○ A clear rationale and justification for the choice of tech stack and technologies on the server-side, front end and back-end development. ○ The main features and functionalities of the project captured in the video presentation clearly and with confident voice.
To obtain a 1st mark (70%), you must have (in addition to the above):	○ A comprehensive review of the functionalities and features with a critical discussion around risk assessment for any missing functionalities from the MVP ○ A video recording capturing the key functionalities and capabilities of the website, stating how the MVP requirements are delivered. ○ Provided a critical discussion around the evolution of the project and how future works can contribute to this. ○ Submitted work meets the essential objectives rand list of requirements in the MVP and delivers advanced functionalities based on a review of the previous works and gaps identified. ○ Excellent written style/ delivery which demonstrates some originality, innovation and insight in the subject area. ○ Demonstrates an excellent range of academic and technical writing skills. ○ The ReadMe file holds comprehensive instructions and guidelines on how to replicate the project structure, install required dependencies and packages, and execute the project. ○ Students' reflections include changes to the project timeline and challenges they have faced in completing the project. ○ A comprehensive discussion around their database design and an ERD diagram of the database ○ Excellent reference to and application of a wide range of relevant reading from a variety of sources and research-informed literature. ○ A comprehensive level of risk assessment and how risk factors will be contained within the scope of the project.

You should note that a scaling mark scheme is employed for the evaluation of your submission/learning. Hence, if you do not meet the pass requirements in any criteria, you may need to resubmit or repeat the module. Ensure that you thoroughly understand the marking criteria and you align your project with them.

HELP AND SUPPORT

- Please use the module handbook and the Computer Science Teams site as a source of information. Do try and find the answer yourself before reaching out for help.
- Support will be provided via Microsoft Teams and email. You can also ask questions during your timetabled sessions. You may request a one-to-one meeting with a tutor during their published office hours.
- The Student Support team are available seven days a week to support you in all aspects of Student life. This could be for support relating to your course, your accommodation or for more general advice such as relationships or your wellbeing. Log in to the LTU app to access support services.
- The full set of university guidelines on assessments, deadlines, and extensions is available on the Taught Programme Academic Regulations (pages 12-15).
- You can assessment regulations in your MyLTU app.