|  |  |
| --- | --- |
| **Journal Submission 5**  **Student name: Kawthar Mohammad Adam**  **Student email address: km2065@hw.ac.uk**  **Date of submission: 21/10/2024** | |
| **Title of dissertation and brief description** | Coursework Submission Deadline Visualiser System. A website for helping students submit their coursework on time. It will have a progress meter to visualise the progress, something like a project management tool. |
| **Communicating with supervisor** | * 4/10/24 (Teams): Front end review and testing discussion (I)   + Assumptions:     - Does not support group projects     - Deadline submission is before midnight (GMT +4)   + Github is not secure for backups, can crash so have other local backups   + The final report must have a revised literature review, methodology and evaluation. Testing is crucial   + Mention student or instructor dashboard   + Keep the colours of the courses same (timeline and dashboard), make the dashboard coursework colour light grey and only the border colour different   + coursework progress bar colour does not match with a darker shade of the tabs   + competitor application coursework (heading must mention that it is a coursework)   + board renamed progress board   + Students only focus on deadlines not on the process. this tool will help them in work life since it is a mix of lms with project management tool (kanban)   + Justify and reflect how your tool evolved   + use journals to show progress   + mention all issues   + include the approach, agile solo, considering supervisor as a client   + screenshots in appendix * 7/10/24 (Teams): Front end review and testing discussion (II)   + Make the borders thicker for the student dashboard boxes   + Leave the sharing part   + Add all students in one course   + Testing for students on 21st October   + 10 students   + 5 professors   + For students, print consent, 2 questionnaires, and task sheet   + Ask if screen recording is allowed   + All testing I did should be there in the report. The ones I have done so far during development. Consult GitHub and make those test cases. The failed ones and passed ones. Justify failing   + One questionnaire will be SUS, and the second will be specific asking them how important this feature is, etc... to get feedback   + 29th Nov is the submission   + Supervisor will set me up with 4th year undergraduate students attending a lecture on Monday, 21st of October at 5 PM. Testing might continue the next day if I don’t get enough participants on the first day * 18/10/24 (Teams): Final student portal review (front-end, back-end, and testing strategy) before testing   + Outline for the progress bar, call it work progress, and show the percentage on the bar   + Remove bar from view coursework   + Guide students only to functional pages   + Don’t keep more than 3 open-ended questions   + Shift questions to Likert scale   + Both questionnaire and survey can be online   + Can screen record   + Estimate time for testing (may be 30 mins) |
| **References consulted** | * YouTube: Tutorials on Next * Documentation, ChatGPT, and stackoverflow: Help with coding and debugging * Figma: Design * LTDI pdf (Canvas): To get information about evaluation and system log data (screen recording for testing) * <https://marker.io/blog/usability-testing-template#website-homepage-usability-template>: For usability testing documents like questionnaire and task sheets * <https://testfort.com/blog/why-your-project-needs-ui-ux-testing>: testing checklists * <https://medium.com/@userfocus/the-1-page-usability-test-plan-dbc8c3d7fb54>: usability plan template |
| **Tools**  **explored/used** | * Jira: For Scrum * GitHub: For version control and backup * Iconscout: For free icons (smiley faces) * Figma: Design * <https://www.freeconvert.com/png-to-ico/download>: To get favicon (png to ico) * Postman: To test post APIs * MySQL Workbench: Creating database and ERD * Microsoft forms: To create SUS * Ms Word: Developing the testing documents * Vs Code: For programming * Google Chrome: For viewing the website * Command prompt/PowerShell: For installing libraries |
| **Other work carried out** | * Completed the front-end for the whole website * Created ERD * Created database * Completed student backend * Created SUS survey * Created information sheet * Created task sheets for students * Created consent form * Created post-usability questionnaire * Created usability plan for myself * Recruited students for testing * Student testing |
| **Plan the next 2 to 3 weeks** | * Complete staff portal * Fix literature review * Contact professors for testing and send invites * Prepare testing documents for staff testing * Consult previous journals and Git commits to prepare final report (especially the testing phase which includes unit and integration). Also mention all struggles and issues faced and all new skills that needed to be acquired in order to complete the project |
| **Overall Reflection\n** | It was difficult since I have never worked on backend and APIs. Some parts of the website are hardcoded due to time restrictions and increased complexity. Luckily, next js can be used for front and backend. I was afraid at first because of how there aren’t many tutorials on Next js 14 but it went fine. Especially, it saved me from learning node js and redux. That would have taken much longer I believe.  The database part went smoothly too, but I need to make changes to the ERD because during development, I had to modify the database, add some columns, change datatypes (file attachment was blob but then I changed it to varchar 100). However, I did make a few mistakes. I was using npm run dev the whole time since my focus was on development. The website is slower in development mode. Upon researching, I found out on the 17th of October that I can use npm run build and npm start to load my website in production mode. However, when I tried it, there were several warnings and errors regarding issues that needed to be resolved in order to build the app. This would take very long so I decided to proceed with npm run dev for testing. It will be slow but I can’t take major risks at this time. If I would have known earlier, I could have fixed those issues earlier during development. Making changes now will be very time consuming and the kanban API needs to be changed. Nope, can’t take the risk.  The most difficult part for me was the kanban board and timeline. They took really long. Also, post is harder than get. That also took long because I did post at the end when I wanted to update the status, etc… It took me some time to get a hold of the backend like passing parameters and all.  I also added tailwind for help but that backfired because I added it almost at the end of the front-end development and tailwind has its own default styles. This completely changed my website’s styling so then I had to remove it from my website code. It is still there in the config file but I’m not using it. |
| **Additional section** | Overall:   * Need to complete staff portal before October ends * Staff testing also needs to be completed immediately after the staff portal is ready * I made mistakes because I didn’t research in depth. I jumped into the implementation without proper research on technologies and their setups. I found out too late about them. Having them earlier could have made my work easier * It was difficult and I regret not starting earlier but this is what my projects aims to change * Hardcoding exists but my focus is on delivering a usable design incorporated with time and project management functionalities to help students with their academics and prepare them for their professional lives |