# Exercise 1 Network Diagram and Critical Path Analysis

1.

Et bilde som inneholder diagram, sketch, plan, Teknisk tegning

Automatisk generert beskrivelse

2.

ES, EF, LS, LF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Project Activities | Earliest Start | Earliest Finish | Latest Start | Latest Finish |
| A | Market Research | 1/31/24 | 2/14/2024 | 1/31/24 | 2/14/2024 |
| B | Requirement Analysis | 2/15/2024 | 2/22/2024 | 2/15/2024 | 2/22/2024 |
| C | UI/UX Design | 2/23/2024 | 3/15/2024 | 2/23/2024 | 3/15/2024 |
| D | Backend Development | 2/15/2024 | 3/14/2024 | 3/8/2024 | 4/5/2024 |
| E | Database Setup | 2/23/2024 | 3/8/2024 | 3/22/2024 | 4/5/2024 |
| F | API Integration | 3/15/2024 | 3/29/2024 | 4/8/2024 | 4/22/2024 |
| G | Frontend Development | 3/18/2024 | 4/22/2024 | 3/18/2024 | 4/22/2024 |
| H | Testing Phase 1 | 4/23/2024 | 4/30/2024 | 4/23/2024 | 4/30/2024 |
| I | User Acceptance Testing | 5/1/2024 | 5/15/2024 | 5/1/2024 | 5/15/2024 |
| J | Final Revision | 5/16/2024 | 5/23/2024 | 5/16/2024 | 5/23/2024 |
| K | Launch Preparation | 5/24/2024 | 5/31/2024 | 5/24/2024 | 5/31/2024 |
| L | App Launch | 6/3/2024 | 6/4/2024 | 6/3/2024 | 6/4/2024 |

3, 4, and 5.

Critical Path: A > B > C > G > H > I > J > K > L

Takes 89 work days, given the 1 day lapse between ending an activity and starting the next one.

Path 2: A > B > E > F > H > I > J > K > L

21 days of slack time, given that E can be done before D, and F before G

Path 3: A > D > F > H > I > J > K > L

17 days of slack time, given that F can be done before G

# Et bilde som inneholder tekst, skjermbilde, line, nummer Automatisk generert beskrivelseExercise 2 Gantt chart

The critical path of this capstone project is 9 weeks.

# Exercise 3

Et bilde som inneholder tekst, diagram, line, Font

Automatisk generert beskrivelse

Disorganized HTML: Continuously test the structure of the website to make sure it’s readable and understandable. Adding «test»/temporary-CSS might make it easier to spot rooms for improvement. This should be done before the website is released, otherwise, it’ll be taken down and fixed; a delay which shouldn’t be necessary. Creating the HTML should be the first thing that gets completed.

Bugs: Test the website and try to check every function and ways of navigating it and look for mistakes. Have a group of testers look through it and deliver reviews and findings. Note down mistakes and fix them one by one, starting with the bugs that affects the most and work the way down.

Unoptimized CSS: Make sure the website is optimized in accordance with standards regarding universal design, in order to avoid complaints about the website. Again, this is a delay which shouldn’t be necessary due to it being front-end programming.

Overcomplicated design: There should not be too much focus on detailing on the website. The website should have an easy-to-navigate look and not be overwhelming / confusing to look at. Simplicity is important here, so it would be useful having a competent graphic designer on the development team.

Inefficient caching: Have an appropriate amount of focus on the website’s back-end programming and keep record of all transactions in a database.

Excessive HTML requests: Add a limit to how many users may enter the website at a time, by organizing a queue. If too many may enter the website at a time it’ll be at risk of trying to conduct too many transactions at once, which could temporarily crash it, which is a very frustrating delay.