COMP3000 Computing Project

2024/2025

Project Title

Teaching and Reinforcing Vital Cybersecurity Skills using OpenAI's ChatGPT

Links

Source code (GitHub repository): https://github.com/oliprev/OP_COMP3000
Trello board: https://trello.com/b/XX2Dc1pz

Project Vision

The aim of this project is to develop a solution targeted at both companies and their workforces, as well as everyday technology users who wish to enhance their cybersecurity skills. The web application is designed to address the growing issue of online vulnerabilities, which can lead to significant social, economic, and even political repercussions for organisations.

To tackle these risks, the proposed solution will be a full-stack web-based application (thus ensuring compatibility across various devices). It will integrate with OpenAI's ChatGPT to provide practical, AI-driven content tailored to the users' needs. There may also be scope for monetisation.

Key features of the application will consist of:

- Dynamic content creation: The AI will generate scenarios and challenges to keep users engaged and aware – with varying levels of difficulty.
- Text-based lessons: There will be a portion of the application dedicated to teaching about different cybersecurity principles – which will be updated regularly to maintain relevance.
- Phishing email simulations: Users will have emails generated for them to practise identifying signs of phishing – with varying levels of difficulty.
- Achievements and statistics: Users can track their progress through achievements and detailed statistics – with incentives for daily use.
- Administrative review: Organisations will have the ability to review user performance of employees to visualise overall improvement.

This project seeks to deliver a challenging, yet engaging educational platform that not only sharpens cybersecurity skills but also offers companies and individuals a way to monitor and improve their overall online security toolkit.

Risk Plan

I think one of the biggest risks is concerning the use of ChatGPT – despite it being a hugely powerful platform, it can also provide false information. Misinformation will do little to benefit any users, so it is up to me to try and diminish the risk of this as much as I can through ensuring queries are succinct. As well as this, I have relatively limited software experience with APIs, so it will certainly be a challenge to establish a link between my application and OpenAI's ChatGPT. Prior to implementation I will practise creating web applications that communicate with API's. Another issue I have may be procrastination due to having depression – however, I am good at seeing past this and looking at the bigger picture. Below, I have inserted a table showing a comprehensive list of events that may hinder me, and their respective risks; it follows Agile Software Development Lifecycle principles.

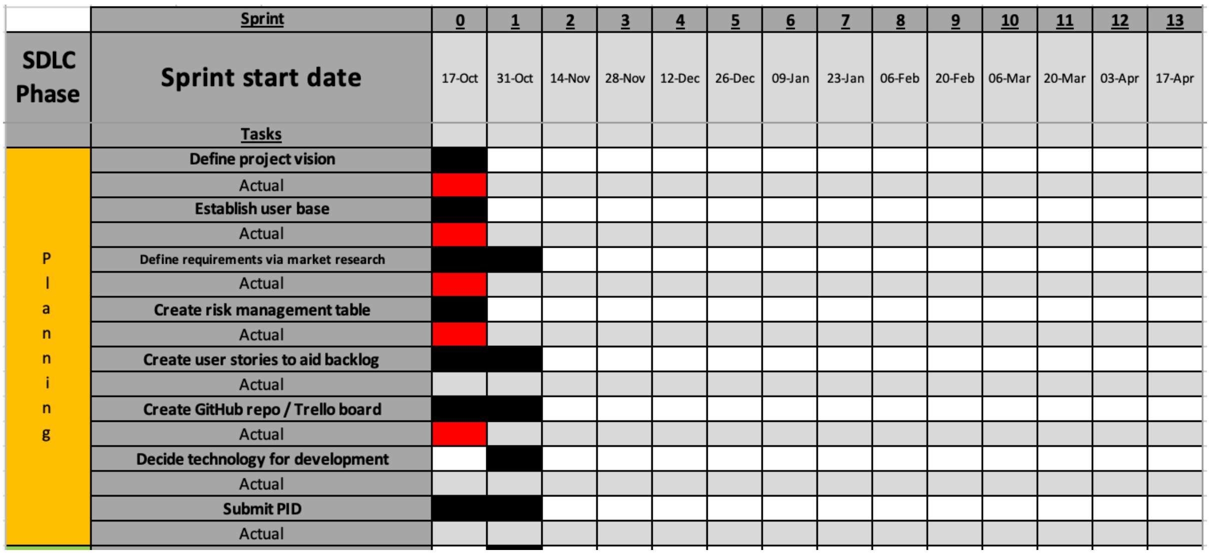
Phase	ID	Event	Likelihood	Impact	Overall risk
Planning	1	Running out of time to implement core features	2	9	18
Planning	2	Underestimating time required to complete tasks or project	4	6	24
Planning	3	Unclear requirements / scope too broad	4	5	20
Planning	4	Lack of communication between supervisor and myself	3	6	18
Planning	5	Poor risk assessment – meaning some unexpected events may crop up and cause delays	5	5	25
Planning	6	Lack of planning for setbacks that may stem from technologies not working as expected	5	7	35
Design	7	Design is too complex	4	8	32
Design	8	Design is inadequate – heavily reducing User Experience (UX)	2	9	18
Design	9	System architecture design is not established correctly – meaning implementation may suffer	3	10	30
Design	10	User stories not expansive or broad enough	4	5	20
Design	11	Poor requirements leading to design rework	4	6	24
Implementation	12	Lack of technical skills leading to poor application quality	3	10	30
Implementation	13	Implementation takes longer than anticipated	5	4	20

Implementation	14	Core components of web application not working as expected	4	9	36
Implementation	15	Coding does not follow best practices, leading to inefficiencies	4	5	20
Implementation	16	Problems concerning project compatibility when it comes to devices that I am coding on	2	7	14
Implementation	17	Extensive coding / crunch, meaning that I may be subject to burnout	6	7	42
Implementation	18	Change in requirements leading to temporal setbacks (scope creep)	6	6	30
Implementation	19	Presence of security vulnerabilities that could jeopardise user safety	4	8	32
Implementation	20	ChatGPT API has an outage, meaning user cannot interact with application as intended	1	10	10
Implementation	21	Integration of different components (API calls, etc.) not working as expected	3	9	27
Testing	22	Lack of testing core components leading to poor functionality	2	7	14
Testing	23	Testing not carried out across different platforms / browsers, leading to potential UX problems	4	6	24
Testing	24	Poor time management, leading to running out of time for thorough testing of all systems	5	5	25
Testing	25	Poor time management when it comes to debugging critical components after failing testing	5	7	35
Testing	26	Dependence on ChatGPT during testing – which may be unavailable	1	9	9
Testing	27	ChatGPT does not provide results as expected	5	9	45
Testing	28	Relying primarily on functional testing as opposed	2	7	14

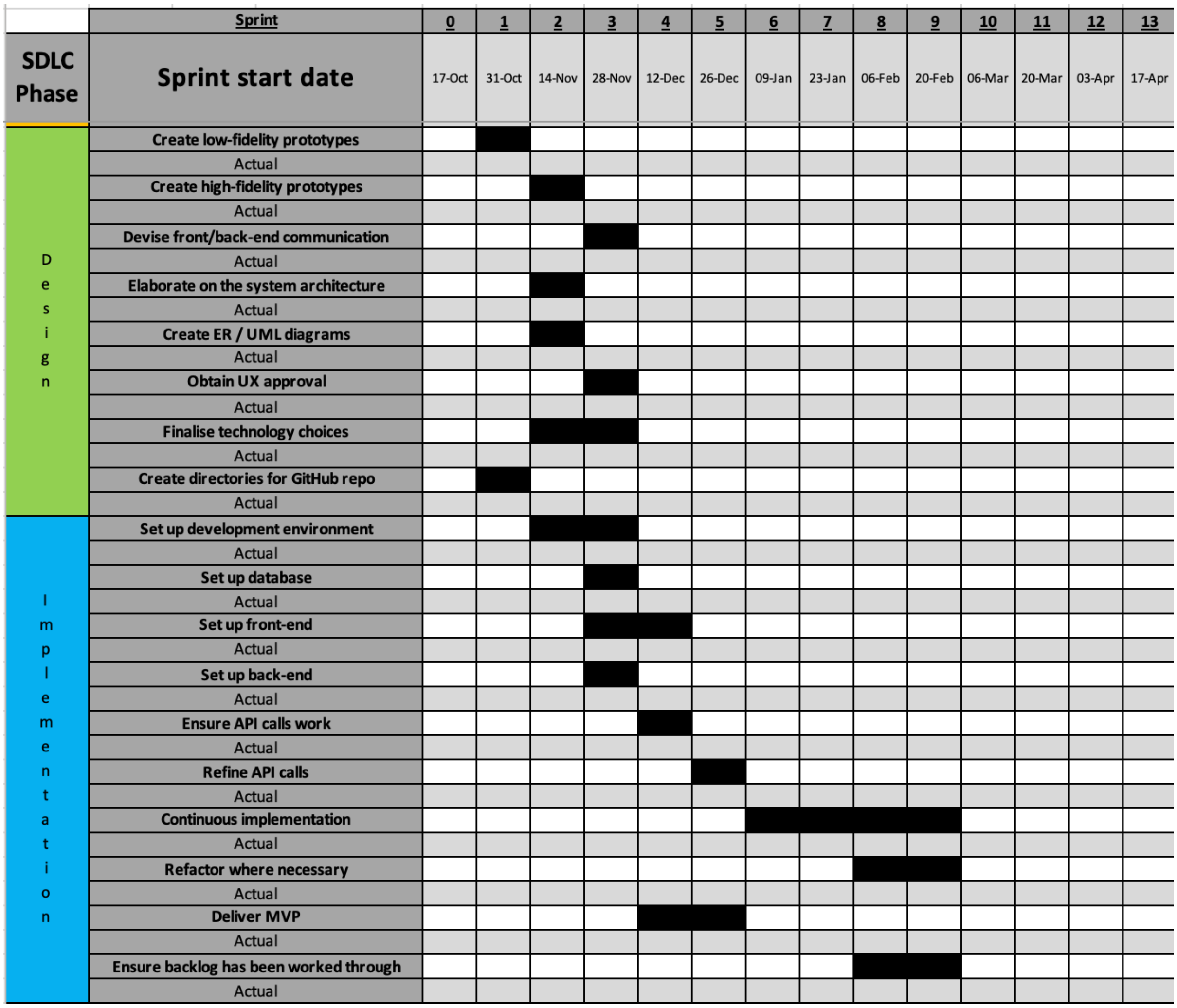
		to more practical testing – meaning UX may be poor			
Testing	29	Lack of testing for security vulnerabilities	6	7	42
Deployment	30	Packaging the project does not work as expected	4	8	32
Deployment	31	Software does not work as expected after deployment	2	9	18
Deployment	32	Lack of Wi-Fi meaning that calls to ChatGPT cannot be made	4	7	28
Deployment	33	ChatGPT outage, meaning calls to API cannot be made	1	9	9
All	34	Becoming ill, impeding progress	6	4	24
All	35	Prioritising inefficiently	3	5	20
All	36	Project files being corrupted	2	10	20
All	37	Supervisor becoming ill – meaning standups / meetings may not go ahead as planned	5	5	25
All	38	Low morale leading to falling behind, delaying project	4	6	24

Proposed Gantt chart

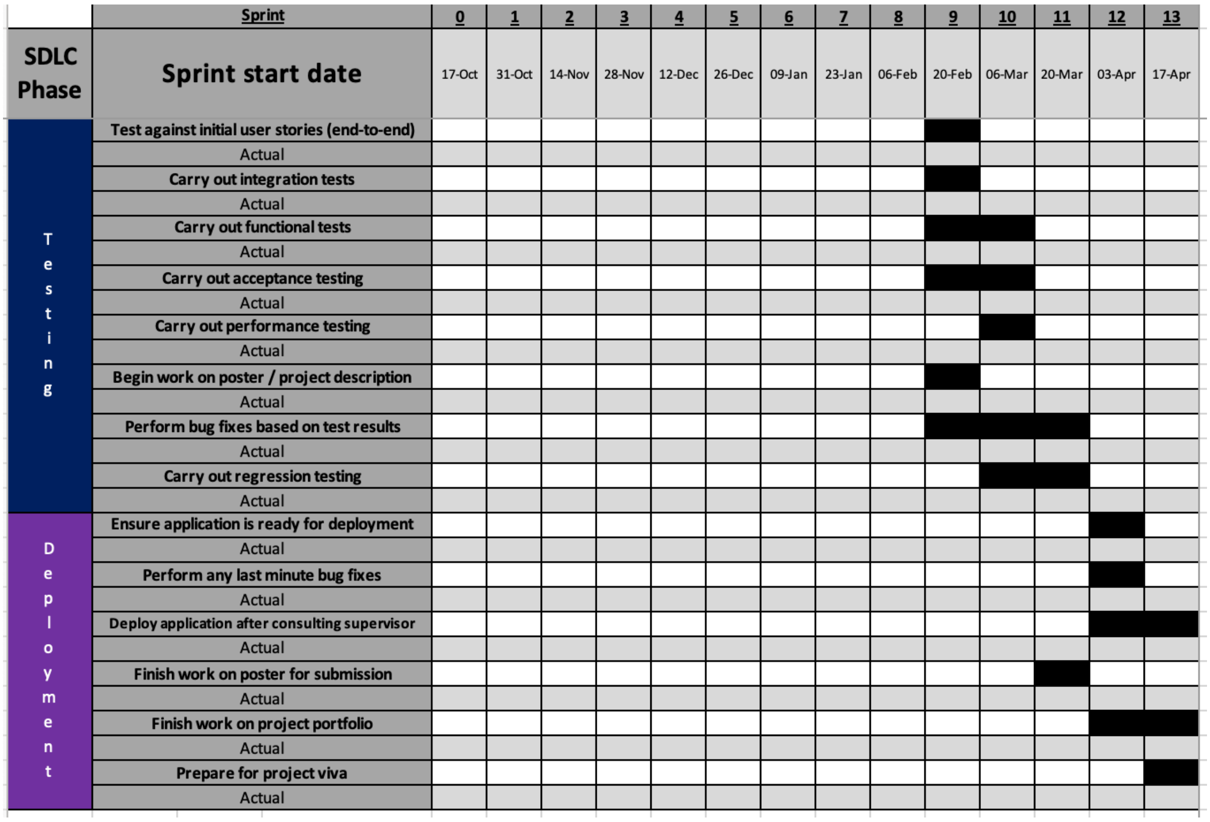
Below, you will find screenshots of my Gantt chart that I developed. It covers all SDLC stages and is a high-level estimate of my project and how it will all fit together. I tried to make it as comprehensive as possible at this early stage. A link to it is also included below if required.
https://liveplymouthac-my.sharepoint.com/:x:/g/personal/oliver_prevett_students_plymouth_ac_uk/EcfqFWAo3lZIieuyZixgAg4B_1_1wVZVTSX8VqcgPEpffw?e=E50Ose



	Sprint	0	1	2	3	4	5	6	7	8	9	10	11	12	13
SDLC Phase	Sprint start date	17-Oct	31-Oct	14-Nov	28-Nov	12-Dec	26-Dec	09-Jan	23-Jan	06-Feb	20-Feb	06-Mar	20-Mar	03-Apr	17-Apr
	Tasks														
Planning	Define project vision	■													
	Actual	🟥													
	Establish user base	■													
	Actual	🟥													
	Define requirements via market research	■	■												
	Actual	🟥													
	Create risk management table	■													
	Actual	🟥													
	Create user stories to aid backlog	■	■												
	Actual														
	Create GitHub repo / Trello board	■	■												
	Actual	🟥													
	Decide technology for development		■												
	Actual														
	Submit PID	■	■												
	Actual														



	Sprint	0	1	2	3	4	5	6	7	8	9	10	11	12	13
SDLC Phase	Sprint start date	17-Oct	31-Oct	14-Nov	28-Nov	12-Dec	26-Dec	09-Jan	23-Jan	06-Feb	20-Feb	06-Mar	20-Mar	03-Apr	17-Apr
Design	Create low-fidelity prototypes		■												
	Actual														
	Create high-fidelity prototypes			■											
	Actual														
	Devise front/back-end communication				■										
	Actual														
	Elaborate on the system architecture			■											
	Actual														
	Create ER / UML diagrams			■											
	Actual														
	Obtain UX approval				■										
	Actual														
	Finalise technology choices			■	■										
	Actual														
	Create directories for GitHub repo		■												
	Actual														
Implementation	Set up development environment			■	■										
	Actual														
	Set up database				■										
	Actual														
	Set up front-end				■	■									
	Actual														
	Set up back-end				■										
	Actual														
	Ensure API calls work					■									
	Actual														
	Refine API calls						■								
	Actual														
	Continuous implementation							■	■	■	■				
	Actual														
	Refactor where necessary									■	■				
	Actual														
	Deliver MVP					■	■								
	Actual														
	Ensure backlog has been worked through									■	■				
	Actual														



	Sprint	0	1	2	3	4	5	6	7	8	9	10	11	12	13
SDLC Phase	Sprint start date	17-Oct	31-Oct	14-Nov	28-Nov	12-Dec	26-Dec	09-Jan	23-Jan	06-Feb	20-Feb	06-Mar	20-Mar	03-Apr	17-Apr
Testing	Test against initial user stories (end-to-end)										■				
	Actual														
	Carry out integration tests										■				
	Actual														
	Carry out functional tests										■	■			
	Actual														
	Carry out acceptance testing										■	■			
	Actual														
	Carry out performance testing											■			
	Actual														
	Begin work on poster / project description										■				
	Actual														
	Perform bug fixes based on test results										■	■	■		
	Actual														
	Carry out regression testing											■	■		
	Actual														
Deployment	Ensure application is ready for deployment													■	
	Actual														
	Perform any last minute bug fixes													■	
	Actual														
	Deploy application after consulting supervisor													■	■
	Actual														
	Finish work on poster for submission												■		
	Actual														
	Finish work on project portfolio													■	■
	Actual														
	Prepare for project viva														■
	Actual														

Keywords

- Cybersecurity
- Web application
- Full stack
- Artificial intelligence
- ChatGPT
- Simulation
- Agile
- Social engineering
- Dynamic content
- Teaching
- APIs
- Awareness
- Visual Studio Code
- MEAN
- MERN