|  |  |  |
| --- | --- | --- |
| **Bachelor of Computing Systems** | | |
| **Course No: ISCG7420** | **Web Application Development** | **Level: 7**  **Credits: 15** |

|  |  |
| --- | --- |
| **Student Name:** | **Student ID:** |
| **Assessment Type: Assignment** | **Weighting: 60%** |
| **Due Date and Time: November 14th 2025 17:00PM** | **Total Marks: 100** |

|  |  |
| --- | --- |
| **Student declaration**  I confirm that:  • This is an original assessment and is entirely my own work.  • The work I am submitting for this assessment is free of plagiarism. I have read and understood the [Academic Integrity Procedure](https://www.unitec.ac.nz/sites/default/files/public/documents/AC%202.8%20Academic%20Integrity%20Procedure_20231025.pdf) (including the key principles about using the use of Generative Artificial Intelligence (GenAI) listed in Section 3.2). I have also read and understood the [Student Disciplinary Statue](https://www.unitec.ac.nz/sites/default/files/public/documents/AC5.0%20Student%20Disciplinary%20Statute%20Aug-2021.pdf).   * Where I have used ideas, tables, diagrams etc of other writers, I have acknowledged the source in every case. | |
| **Students Signature:** | **Date:** |

**Assessment Mapping**

After completing this assessment, the student will have met the following learning outcomes:

|  |  |  |
| --- | --- | --- |
| **Learning Outcomes** | **Task A**  **(30 Marks)** | **Task B**  **(70 Marks)** |
| 1. 1. Discuss the philosophy of client-server computing and its impact to the computing industry. | ✓ |  |
| 1. 5. Design and develop a database client-server solution that meets specified organisational requirements using database and modern data access technologies. |  | ✓ |

**Assessment information:**

* This is an individual assignment.
* Read the provided scenario and complete both tasks.
* Task 1: Reflection on Server-Side vs. Full-Stack Development. Present your findings in a 1000 word report [+/- 10%], excluding reference list.
* Task 2: Design and develop a Full-Stack Development with Django REST Framework and React JS.
* Ensure all sources are referenced in-text and include a complete reference list at the end of the report, following APA 7th edition referencing guidelines.
* If you submit the report and code you did not write, you will fail this assignment. If you need help, ask your lecturer well before the deadline rather than resorting to plagiarism.

**Assessment submission instructions:**

* Complete the cover sheet (page 1 of this document) and attach it to the front of the assessment.
* Submit your Task 1 via the Turnitin link on the Moodle course page.
* For Task 2, use the Assignment 1 directory in your class GitHub repository to store your task and add your lecturer as a collaborator. Your lecturer will periodically clone your repository to monitor your progress. On the assignment due date (and up to 72 hours afterward), they will download the most recent version of your Task 2 submission. Ensure GitHub always contains the latest version of your code.
* For Task 2, if you fail to push your most recent changes by the deadline, your lecturer will assess the latest version available on GitHub. Regularly update your files to ensure your submission is up to date.

**Scenario:**

**Te Whare Rūnanga** **Ltd** is developing a web application to facilitate for reserving conference room reservations. The company has 10 conference rooms, and the application should enable users to view available rooms, make reservations, and manage existing reservations.

1. **View Available Rooms**: Users can browse a list of available conference rooms.
2. **Make Reservations**: Users can book a conference room for a specific date and time.
3. **Manage Reservations**: Users can view, edit, or cancel their existing reservations.
4. **User Authentication**: Users must log in to make or manage reservations.
5. **Admin Panel**: Administrators have the following capabilities:
   1. Manage rooms: Add, Edit, and Delete Conference Rooms details.
   2. Reserve for Users: Make reservations on behalf of users.
   3. Cancel User Reservations: Cancel bookings made by users.
   4. View All Reservations: Oversee and manage all reservations in the system.
   5. Manage User Accounts: Add, edit, or delete user accounts.
6. **Notifications**: Users receive confirmation and reminder notifications for their reservations.

**Task 1: Reflection on Server-Side vs. Full-Stack Development [30 Marks]**

**Instructions:**

* Reflect on the work you completed in Assignment 1 and Assignment 2 (Task 2) for developing the conference room reservation system for the above scenario, with a focus on Server-Side Development and Full-Stack Development.
* Your reflection should analyse your learning, challenges faced, and key takeaways.
* Present your insights in a 1000-word report (±10%), excluding the reference list (APA 7th referencing style). The report should be logical and well-structured, including an introduction, main discussion, conclusion, and proper referencing. (6 mark)
* Report requirements:

1. Definition and Scope (4 Marks)

Explain the development requirements of the server-side development and full-stack development.

1. Technologies used (4 Marks)

Identify and compare technologies used in both server-side and full-stack development.

1. Responsibilities (4 Marks)

Discuss the different responsibilities in server-side development, where the server handles everything, versus full-stack development, where the client hosts the interface and the server only provides data.

1. Development Focus (4 Marks)

Analyse the key focus areas in server-side vs. full-stack development.

1. Skill Set (2 Marks)

Describe the essential skill sets for server-side and full-stack developers.

1. Project Workflow (2 Marks)

Explain the workflow differences between server-side and full-stack development.

1. Career Opportunities (4 Marks)

Provide an overview of career opportunities in both server-side and full-stack development.

**Task 2: Full-Stack Development with Django REST Framework and React [70 Marks]**

**Instruction:**

Design and develop a dynamic web application for a conference room reservation system using Django REST Framework and React.

Requirements:

* Develop the backend using Django REST Framework.
* Build the frontend with React.
* Host the completed application on Vercel, ensuring all required packages are configured.
* Upload your source code to GitHub and maintain version control throughout the development process.

Ensure that your implementation follows best practices in web development, including performance, code quality & structure, version control & deployment, API design, usability and maintainability.

You will demonstrate the dynamic web application developed for a conference room reservation system.

**Assessment Marking Sheet**

Student ID:

|  |  |  |
| --- | --- | --- |
| **Contents/Criteria** | **Marks Allocated** | **Your Marks** |
| **Task 1** |  |  |
| Definition and Scope | 4 marks |  |
| Technologies Used | 4 marks |  |
| Responsibilities | 4 marks |  |
| Development Focus | 4 marks |  |
| Skill Set | 2 marks |  |
| Project Workflow | 2 marks |  |
| Career Opportunities | 4 marks |  |
| Report structure and referencing | 6 marks |  |
| **Task 2** |  |  |
| Application functionality | 15 marks |  |
| Django (Backend) | 15 marks |  |
| React (Frontend) | 15 marks |  |
| Hosting on Vercel | 10 marks |  |
| GitHub Version Control | 5 marks |  |
| Code quality and best practices | 5 marks |  |
| Demonstration | 5 Marks |  |
| **Total marks** | 100 marks |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task 1 Marking Rubric** | | | | |  |
| **Performance Criteria** | **Outstanding** | **Accomplished** | **Capable** | **Requires development** | **Total** |
| Definition and Scope  (4 marks) | Provides a clear, in-depth explanation of server-side and full-stack development with well-supported examples and personal reflection on understanding these concepts. 4 marks | Explains server-side and full-stack development with relevant details but limited reflection on understanding.  3 marks | Basic explanation with minimal detail; lacks personal insights or reflection. 2 marks | Incomplete or unclear explanation with gaps in understanding.  0 -1 marks |  |
| Technologies Used (4 marks) | Identifies and thoroughly compares technologies used in both server-side and full-stack development, including reflection on how these technologies were used in the project. 4 marks | Identifies and compares technologies used in both server-side and full-stack development with some discussion of their role in the project.  3 marks | Mentions relevant technologies used in server-side and full-stack development but provides minimal comparison and lacks personal insight.  2 marks | Technologies are incorrectly identified or lack a meaningful comparison.  0 - 1 mark |  |
| Responsibilities  (4 marks) | Clearly discusses responsibilities in server-side and full-stack development, with personal reflection on challenges and learning.  4 marks | Discusses responsibilities with some examples but minimal personal reflection. 3 marks | Provides a general discussion but lacks depth or personal insights.  2 marks | Fails to clearly explain responsibilities or provides incorrect information.  0 - 1 mark |  |
| Development focus  (4 marks) | Provides an insightful analysis of key focus areas in server-side vs. full-stack development, incorporating reflection on what was learned through hands-on work. 4 marks | Analyses focus areas with reasonable explanations but lacks deeper personal insights. 3 marks | Addresses focus areas but with limited analysis or clarity; little personal reflection. 2 marks | Inadequate or incorrect analysis with missing key points. 0 -1 mark |  |
| Skill Set  (2 marks) | Clearly describes essential skills required for both roles, with reflection on personal skill development during the project. 2 marks | Identifies skills with some discussion of personal learning but lacks depth. 1.5 marks | Mentions some relevant skills but lacks clarity or self-reflection. 1 mark | Provides an incomplete or incorrect list of skills with no reflection.  0 – 0.5 mark |  |
| Project Workflow  (2 marks) | Clearly explains workflow differences between server-side and full-stack development with well-supported reasoning and reflection on challenges encountered in project management. 2 marks | Explains workflow differences between server-side and full-stack development with some reasoning but lacks detailed reflection on experience. 1.5 marks | Provides a basic explanation but lacks examples and personal insights. 1 mark | Limited or unclear explanation of workflow differences.  0 – 0.5 mark |  |
| Career Opportunities  (4 marks) | Provides a well-structured overview of career opportunities in server-side and full-stack development, with reflection on how the experience has shaped career interests. 4 marks | Discusses career opportunities in server-side and full-stack development but lacks strong personal connection. 3 marks | Mentions career opportunities server-side and full-stack development but lacks depth or insight into personal aspirations.  2 marks | Minimal or unclear discussion of career opportunities.  0 – 1 mark |  |
| Report  (6 marks) | Report is well-structured and clear with logical flow. Writing is free from major grammar or spelling errors. Referencing is correctly formatted and follows the required citation style.  5 – 6 marks | Report is mostly well-organised and clear, with minor formatting issues. Some grammar or spelling errors. Referencing is mostly correct but may have minor formatting inconsistencies.  4 marks | Report is somewhat disorganised with noticeable grammar, spelling, or formatting issues. Referencing is attempted but contains multiple errors.  3 marks | Report lacks structure and clarity, with significant grammar, spelling, or formatting errors. Referencing is missing or incorrect.  0 – 2 marks |  |
|  | | | | **Total** |  |
| **Feedback** |  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task 2 Marking Rubric** | | | | |  |
| **Performance Criteria** | **Outstanding** | **Accomplished** | **Capable** | **Requires development** | **Total** |
| Application functionality  (15 marks) | The application is fully functional, meeting all requirements with a user-friendly experience. All core features (room viewing, booking, managing reservations) work as expected, with minor errors or issues.  13 – 15 marks | The application meets most requirements with minor issues. Some features may have small bugs but do not significantly impact functionality.  10 -12 marks | The application includes basic functionality but lacks completeness. Some features may not work correctly or are missing.  7 – 9 marks | The application is largely incomplete, with major missing or non-functional features. 0 – 6 marks |  |
| Django (Backend)  (15 marks) | Well-structured backend with properly implemented RESTful API endpoints (serializers and implements views/viewsets). Token authentication and proper error management.  13 – 15 marks | Mostly functional backend with minor issues in API design, authentication, or error handling. 10 - 12 marks | Basic backend functionality is present but lacks efficiency or security features. API endpoints may have inconsistencies.  7 – 9 marks | Backend is incomplete, poorly structured, or non-functional, with missing key features.  0 - 6 marks |  |
| React (Frontend)  (15 marks) | Well-designed, fully responsive UI with intuitive navigation and data connection through Axios. Components are well-structured with effective state management. Some minor errors in efficiency of data use.  13 – 15 marks | Functional UI with good responsiveness and usability, but some minor UI/UX issues exist or data connection through Axios have issues. React components are used effectively.  10 – 12 marks | UI is functional but lacks polish. Navigation may be unclear or data connection may not be successful, and some components are poorly structured.  7 – 9 marks | UI is incomplete, not user-friendly, or has major functionality issues. Poor navigation and component structure. 0 – 6 marks |  |
| Hosting on Vercel  (10 marks) | Django REST Framework backend and React frontend is successfully deployed on Vercel with all required dependencies and configurations. Runs smoothly with no errors. There may be some package management issues.  9 – 10 marks | Django REST Framework backend and React frontend is deployed and mostly functional, though minor issues exist with configurations or performance.  7 – 8 marks | Django REST Framework backend and React frontend is deployed but one of them has issues. Some configurations may be missing.  4 – 6 marks | Application is not successfully deployed or has major functionality issues making it inaccessible. 0 – 4 marks |  |
| GitHub Version Control  (5 marks) | Code is well-managed using GitHub with a clear commit history, proper branching, and regular updates. 5 marks | GitHub is used appropriately, but commit messages or version control practices could be improved. 4 marks | GitHub repository exists, but commits are infrequent or lack meaningful descriptions.  3 marks | Poor or missing version control; repository is incomplete or lacks updates.  0 – 2 marks |  |
| Code Quality and Best Practices  (5 marks) | Code follows best practices, including clean structure, readability, and maintainability. Application is user-friendly and well-documented. 5 marks | Good coding practices are followed, but some areas could be improved for better maintainability.  4 marks | Some coding best practices are applied, but code readability and usability could be improved.  3 marks | Poor coding practices, lack of readability, and little consideration for usability and maintainability.  0 – 2 marks |  |
| Demonstration  (5 marks) | Delivers an excellent demonstration, highlighting key features, development process, and challenges faced.  5 marks | Delivers a good demonstration, highlighting most key features and development process.  4 marks | Delivers a basic demonstration, highlighting some key features and development process.  3 marks | Fails to prepare or deliver an adequate demonstration.  0 – 2 marks |  |
|  | | | | **Total** |  |
| **Feedback** |  | | | | |