**Criterion A: Planning**

**Defining the problem statement**

I have been visiting my dentist, who is also my neighbour, for over a year now. Unfortunately, the clinic he works in still lacks the latest infrastructure and it is very evident that he has a tight and small budget for improving his machines. He emails himself all the data and teeth pictures of his clients and often complains to me and my dad about how inefficient and time consuming it is to search and find the client’s name in his mail. He doesn’t have many assistants and maintains all the appointment times and dates in an old book which he fills up by himself. After listening to all of this, I decided to make an application that would contain the data and information of his patients in a clean interface and also regularly maintain and update appointment times and fees.

**The Rationale for the Proposed Solution**

I have decided to make a web-based application for this purpose and take all the time needed to ensure it’s made carefully without problems. This is because the user would just need a device with a proper internet connection to access the application from any location and any device with a web browser. This would be very efficient as my dentist would need to run the application on multiple devices in his office to get different data from different machines that are connected to their own unique desktops. It is also very easy to update the web application and prevents the need for the user to separately update each time, thus making its maintenance simple.

I have decided to use Python over Java for my application due to its extensive ecosystem of web frameworks, such as Flask, that provide several comprehensive libraries. Additionally, Python's simple syntax and dynamic typing make it easier to write and maintain code, reducing the overall development time and complexity compared to Java. I have decided to use Flask-SQLAlchemy for the database to store all the user and patient data. Since, I am building a Flask web application, using Flask-SQLAlchemy would allow for a proper integration with it by automating session management. I am also using HTML and CSS for designing the web-pages to give a comfortable look.

Word Count: 363

**Success Criteria**

1. Client should be able to create an account with unique credentials.
2. The application must validate user signup with two-factor email authentication.
3. Client should be able to add new patients.
4. The application must display all the patients in a separate page.
5. Client will be able to filter and search for individual patients based on name, age, or gender.
6. Client should be able to add additional diagnosis for each patient.
7. Client should be able to upload pictures for each patient.
8. Client should be able to view pictures for the corresponding patient as a carousel.
9. Client should be able to add or remove an appointment for the patients.
10. Client should be able to update the appointment time for the patients.
11. The application must not allow for any overlap in timing among the existing and future appointments.
12. Display the grand total number of patients and appointments on the dashboard.
13. Display the appointments of the current day on the dashboard.