Question 2

1. MVP

Full stack AI powered customer portal – since this is an MVP and must be prototyped at the earliest (24 Hrs.), this would be my architecture:

AI Reponses

ML XgBoost Classifier trained on medium to large data

Back-end

Low-code

Node.js with a mock database

Front-end

No-Code

Bubble.io

Explanation:

1. Front end – using a simple drag and drop tool like Bubble.io to design a website/UI that takes in input and forwarded to a node.js server. Alternatively, a simple html/css script can be used to deign a UI with the help of LLM API Calls (provided it is within the budget)
2. Back end – 3 Node.js files. One for running server and setting configurations, one for creating a mockDB of users and ID to ensure security while logging in and accessing account balance. This account balance can be used by the AI-response along with stored credit score to access loan eligibility criteria.
3. AI – Response – The last Node.js life is a XGBoost Classifier trained on credit, account balance, previous loans, existing loans, income etc. data that accesses loan eligibility.
4. Simple API with python is attached as a code in the folder.
5. AI / Low-code
6. AI-tools – I would make use of Cursors free trial to develop an MVP within 24 hrs. With the right prompt engineering, I will ask Cursor to develop a simple front end with HTML and CSS, back end with Node.js and AI-Reponses as a separate file. I will ask Cursor to develop a dependency tree, make interfaces secure and containerize it with docker.

I will make use of Napkin.ai to build workflows and UI to explain product to investors if any.

1. Low-code – Essentially, I would use a low-code platform to only build the front end and/or a mock database. I would use bubble.io, since I have experience using it. This will reduce a bit of time (although not a lot). I will use a low-code free tier of snowflake to build a database and integrate it with back-end. Snowflake integration takes a bit of time since it must be configured with AWS or GCP, but is secure and robust in terms of SQL operations. Any mistakes/queries can be handled on snowflake.