

BUDGET BOSS TECH REQUIREMENTS
Group T108

Adam Simcoe
Christian Do
Nhan Tran
Nhu Ly
Trang Nguyen

PART II -- Technology Requirements (as a Word document or a pdf) (max. 4 pages)

A) Technological Requirements

<u>Category</u>	<u>Technology</u>	<u>Description</u>	<u>Pros</u>	<u>Cons</u>
Frontend Framework	React Native	A framework for building mobile apps for both iOS and Android using JavaScript and React.	<ul style="list-style-type: none"> - Cross-platform (works for both iOS and Android) will allow faster development time. - Large selection of libraries and tools associated with it will increase overall use efficiency. - Supports hot reloading, allowing faster development of features. - Highly compatible with other systems. 	<ul style="list-style-type: none"> - Performance might be slower than native apps. - Requires learning if unfamiliar with React or React Native. - Depends on third party libraries, which could become defunct over time.
Backend Framework	Flask	A Python web framework for building APIs and web applications.	<ul style="list-style-type: none"> - Lightweight and easy to set up. - Flexible for small to medium-sized apps. - Only requires what is necessary to function, very minimalistic in nature. - Uses a very user-friendly and easy to learn language. 	<ul style="list-style-type: none"> - May not be the best option for scaling in the future. - Will require additional tools if the application becomes larger than the current proposed implementation.
Database	SQLite	A lightweight relational database that works well with small applications.	<ul style="list-style-type: none"> - Easy to set up and use. - Comes built in within many environments. - Very resource friendly, requires minimal resources to run efficiently. - Serverless, very portable in nature. 	<ul style="list-style-type: none"> - Not suitable for very large apps or high traffic. - Limited scalability. - Does not support all features other databases have.
HTTP Client	Axios	A JavaScript library used for making HTTP requests from React Native to the Flask backend.	<ul style="list-style-type: none"> - Easy to use for making API requests. - Supports promises and async/await. - Has a large community support. 	<ul style="list-style-type: none"> - May not be necessary depending on direction of development. - May require dependency management to ensure correct version compatibility.
Middleware	Flask-CORS	A Flask extension for handling Cross-Origin Resource Sharing (CORS), allowing React Native to talk to Flask.	<ul style="list-style-type: none"> - Necessary for development to allow communication between React Native and Flask. - Simple to set up and configure. - Reduces development time for handling CORS. 	<ul style="list-style-type: none"> - Improper configurations can introduce security risks if not handled properly. - Adds dependency on the backend. - Over-reliance may lead to more lenient CORS policies.
Graphing Library	Matplotlib	Python library used for creating static and interactive visualizations of data, such as charts and graphs.	<ul style="list-style-type: none"> - Customizable for various different charts and graph types. - Smooth integration with other Python libraries. - Generates visualizations server-side. 	<ul style="list-style-type: none"> - Learning curve if unfamiliar with the library.
Graphing Library	Charts.js	JavaScript library used for creating static and interactive visualizations of data, such as charts and graphs.	<ul style="list-style-type: none"> - Simple and easy to learn. - Supports a large variety of different charts and graphs. - Lightweight, extremely fast rendering. 	<ul style="list-style-type: none"> - Client-side based, not ideal if server-side visualizations are necessary. - Requires WebView to function in React Native.

Authenticat ion	Flask-Login	Flask extension used for managing user authentication and session handling.	<ul style="list-style-type: none"> - Simple to use and implement. - Integrates directly with Flask. - Provides session management for any logged in users. - Lightweight in nature. - Well documented. 	<ul style="list-style-type: none"> - Requires additional setup if adding multi-factor authentication or token based login. - Limited features compared to other frameworks.
Password Encryption	Bcrypt	Python library used for password hashing and verification.	<ul style="list-style-type: none"> - Secure and well documented. - Industry standard and highly regarded. - Uses salting to protect against brute force methods. - Smooth integration with Flask and other Python backends. 	<ul style="list-style-type: none"> - Slower hashing speed - May need additional library dependencies for effective management of user credentials (such as Flask-Login).

B) Learning Plan

Team Member	Responsibility	Technology	Explanation	Existing Skill Level (%)	Explanation	Learning Plan	Resource
Nhan Tran (CTO)	Oversees the technical architecture, backend design, and integration of frontend and backend. Ensures scalability, performance, and resolves technical challenges.	Flask	Flask is essential for backend API development and integrating with the frontend.	40%	Basic Python knowledge; needs Flask routing, middleware, and API skills.	Week 1-4	Flask Mega-Tutorial, Official Docs.
		Flask-CORS	Flask-CORS is needed for managing cross-origin requests, enabling frontend-backend communication.	30%	No prior experience with CORS; needs to handle cross-origin requests.	Week 5-6	Flask-CORS Documentation.
		React Native	React Native is needed to oversee the frontend structure and ensure smooth development and integration with backend APIs.	20%	Basic understanding of mobile app development; needs to deepen understanding of components, navigation, and state management.	Week 7-9	Udemy React Native Advanced Course, React Native Docs.
Adam Simcoe (Marketing)	Designs the user interface and manages frontend development, ensuring user-friendliness and brand alignment. Handles app promotion and user acquisition.	React Native	React Native is essential for developing the app's UI/UX, which is critical for user engagement and overall experience.	35%	Familiar with design principles but new to development; needs hands-on skills.	Week 2-7	Udemy React Native Beginner course.
		Axios	Axios will be used to handle HTTP requests between the frontend and backend, enabling data communication.	25%	Limited HTTP client knowledge; needs to learn API calls using Axios.	Week 8-9	Axios Docs, TutorialsPoint.

Trang Nguyen (Labour Relations)	Develops the backend API, builds server-side logic, and integrates labour relations features like payroll and employee data. Ensures effective communication between systems.	Flask	Flask is required for backend API creation, routing, and integration of features like payroll and employee management.	30%	Basic Python exposure; requires Flask skills for backend development.	Week 2-6	FreeCodeCamp, Flask Tutorials.
		SQLite	SQLite helps manage data storage for employee information, payroll, and other business-related data.	20%	New to databases; must learn schema design and SQL basics.	Week 7-9	w3schools SQLite Tutorial.
		Axios	Axios will be used for making API requests between the backend and frontend, ensuring smooth communication.	20%	Needs to learn how to implement HTTP requests for API calls.	Week 8-9	Axios Docs, Tutorials.
Nhu Ly (Finance)	Manages data analysis, financial calculations, and database design. Works on budget tracking, expense calculation, and financial reporting.	SQLite	SQLite is necessary for storing and managing financial data such as expenses, budgets, and transactions.	40%	Familiar with database concepts; needs practice with SQLite queries.	Week 1-4	Khan Academy SQL course.
		Charts.js	Charts.js will be used to create interactive visualizations of financial data to help users understand their financial status.	25%	Some JavaScript experience; needs exposure to creating visualizations.	Week 5-8	Charts.js Docs, Tutorials.
		Matplotlib	Matplotlib will be needed for advanced visualizations and data analytics, especially for more complex financial graphs and reports.	15%	Little experience with Python libraries; focus on creating graphs for financial data.	Week 9-1	TutorialsPoint, Python guides.
Christian Do (Production)	Oversees feature integration and app deployment. Manages testing, troubleshooting, and ensures smooth production. Focuses on	React Native	React Native is needed for integration, feature testing, and ensuring app functionality.	30%	Basic knowledge of JavaScript; needs to learn component-based architecture.	Week 1-4	React Native Docs, Codecademy.

	data visualization for decision-making.	Flask	Flask will be used for backend feature integration and testing, ensuring the app works correctly in the production environment.	10%	Basic knowledge of Flask; needs to focus on debugging and deployment.	Week 5-9	Flask Debugging Guide, Deployment Tutorials.
		SQLite	SQLite is crucial for managing production data, especially related to the app's performance and real-time user interaction.	30%	Familiar with SQL; needs to troubleshoot production-related database issues.	Week 1-5	SQL for Developers, SQLite Docs.