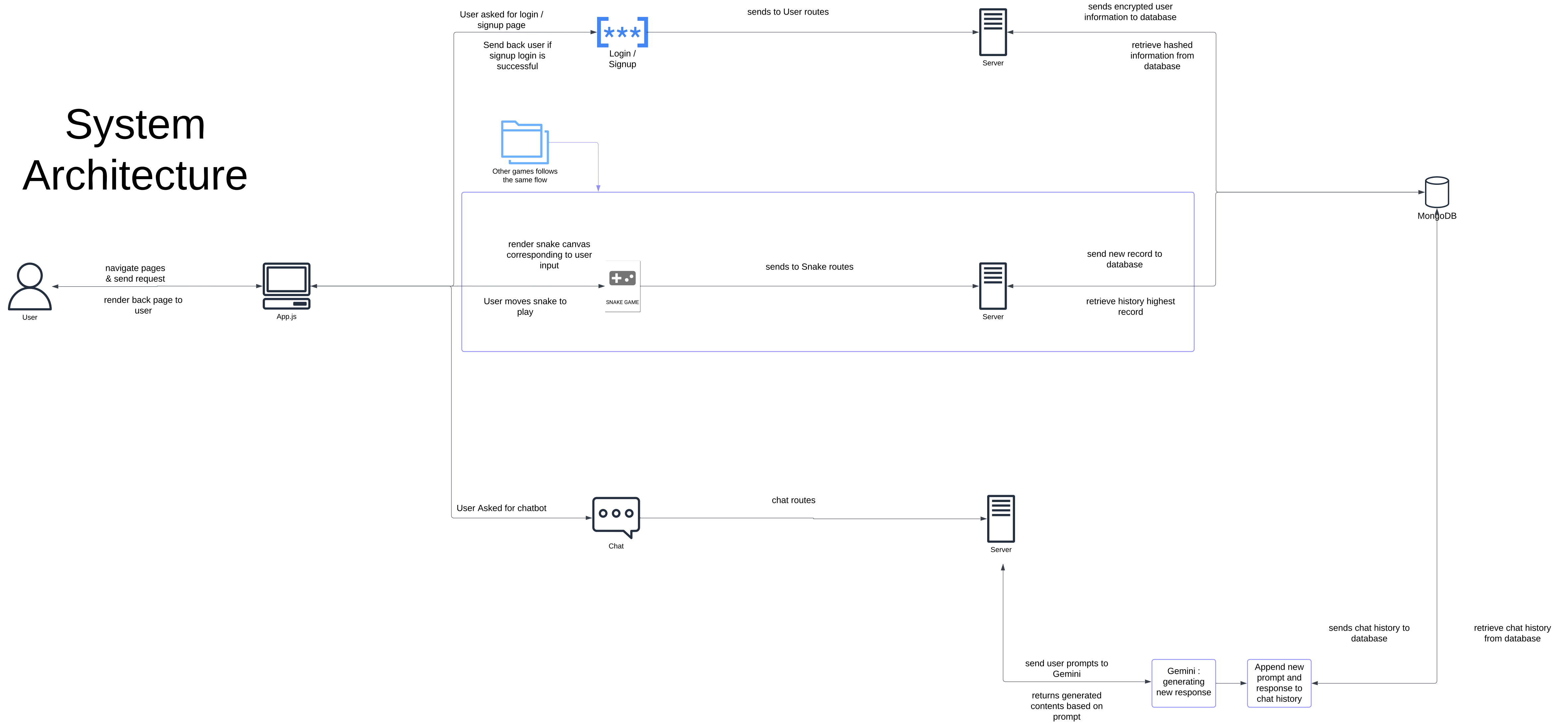


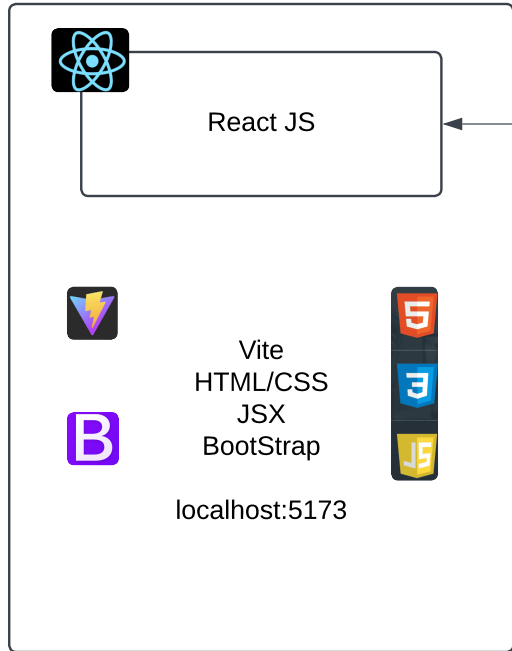
System Architecture



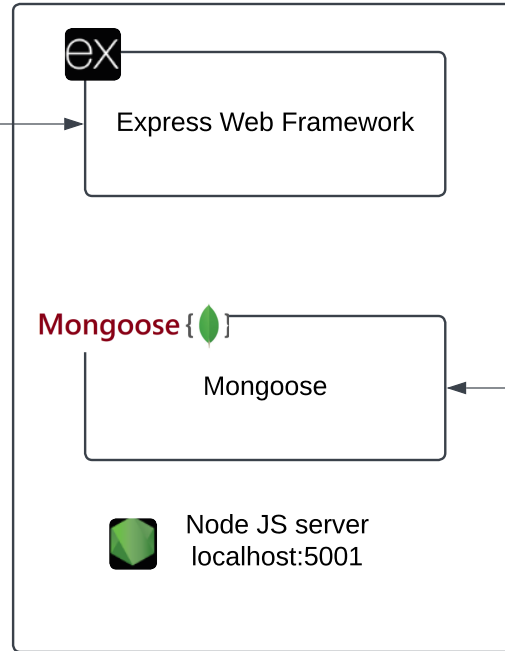
client



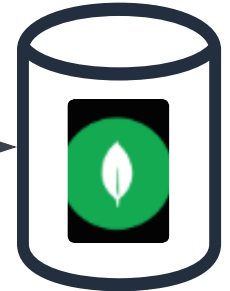
Server



Frontend development



Backend development



MongoDB

Database Management

Class	AboutUs.jsx
Responsibilities	<ul style="list-style-type: none"> - Render an "About Us" section with content. - Provide a structured view containing a heading and a paragraph to display information about the team.
Collaborators	<ul style="list-style-type: none"> - App.js

Class	ChatBot.jsx
Responsibilities	<ul style="list-style-type: none"> - Render the main chatbot interface by encapsulating ChatContainer. - Pass user-specific data (userEmail) as props to ChatContainer.
Collaborators	<ul style="list-style-type: none"> - App.js - ChatBot.css - Database

Class	Dashboard.jsx
Responsibilities	<ul style="list-style-type: none"> - Render user profile information, achievements, and game statistics. - Manage and display the profile image with upload functionality. - Provide a logout function, allowing the user to reset their session.
Collaborators	<ul style="list-style-type: none"> - App.js - Dashboard.css - Database

Class	GameList.jsx
--------------	--------------

Responsibilities	<ul style="list-style-type: none"> - Render a list of games based on <code>gameListObject</code> prop. - Display the list title from <code>listType</code>. - Provide navigation links for each game item.
Collaborators	<ul style="list-style-type: none"> - App.js - GameList.css

Class	GamePage.jsx
Responsibilities	<ul style="list-style-type: none"> - Render the game interface with <code>GameHeader</code>, <code>GameBoard</code>, and <code>GameLoader</code>. - Pass relevant game data and user information to child components. - Log the received data and processed user details for debugging purposes.
Collaborators	<ul style="list-style-type: none"> - App.js - GamePage.css - GameHeader - GameBoard - GameLoader

Class	HomePage.jsx
Responsibilities	<ul style="list-style-type: none"> - Render a three-column layout with placeholder content for each section. - Provide a basic responsive structure using Bootstrap classes.
Collaborators	<ul style="list-style-type: none"> - App.js - Footer component

Class	LoginPage.jsx
--------------	---------------

Responsibilities	<ul style="list-style-type: none">- Render login and registration forms with conditional switching between them.- Handle user login and registration through API calls and manage login state.- Conditionally render the Dashboard component upon successful login.
Collaborators	<ul style="list-style-type: none">- App.js- LoginPage.css- Backend API

CLASS NAME: Footer.jsx

Parent Classes (if any): App.jsx

Subclasses (if any):

Responsibilities:

Generate Footer section of webpage

Collaboraters:

AboutUs.jsxChatBox.jsxDashboard.jsxGameList.jsxGamePage.jsxHomePage.jsxLoginPage.jsx

CLASS NAME: GameLoader.jsx

Parent Classes (if any): GameBoard.jsx

Subclasses (if any): Snake

Responsibilities:

Load and display content based on value it recieved in gameBody

Collaboraters:

SnakeGameWordGameAimTrainerImageMemoryGameSlotGame

CLASS NAME: GameBoard.jsx

Parent Classes (if any): GamePage.jsx

Subclasses (if any):

Responsibilities:

Central layout for game area, displays game interface

Collaboraters:

GameLoaderHelpfulTipsPreviousAttempts

CLASS NAME: HelpfulTips.jsx

Parent Classes (if any): GameBoard.jsx

Subclasses (if any):

Responsibilities:

Display guidance and tips to assist players

Collaboraters:

SnakeGameWordGameAimTrainerImageMemoryGameSlotGame

CLASS NAME: GameHeader.jsx

Parent Classes (if any): GamePage.jsx

Subclasses (if any):

Responsibilities:

Display name of game as a heading in game interface

Collaboraters:

SnakeGameWordGameAimTrainerImageMemoryGameSlotGame

CLASS NAME: Navbar.jsx

Parent Classes (if any): App.jsx

Subclasses (if any):

Responsibilities:

Navigate links in the application for users to access different sections.

Collaboraters:

App.jsxAiChatbotGameListUserLoginIn/OutDashboard

CLASS NAME: PreviousAttempts.jsx

Parent Classes (if any):

Subclasses (if any):

Responsibilities:

Allows users to check their past attempt

Collaboraters:

SnakeGameWordGameAimTrainerImageMemoryGameSlotGameUser

ChatModel.jsx	
Parent class: Subclass:	Collaborator: ChatModel.jsx ChatRoute.jsx Server.js ChatController.jsx
Responsibilities: Creating schema for chat messageSchema chatSchema	

ChatContainer.jsx	
Parent class: Chatbot.jsx Subclass: Message.jsx	Collaborator: ChatModel.jsx ChatRoute.jsx Server.js ChatController.jsx
Responsibilities: Accept user inputs	

Chatbot.jsx	
Parent class: app.jsx Subclass: ChatContainer.jsx	Collaborator: ChatModel.jsx ChatRoute.jsx Server.js ChatContainer.jsx ChatController.jsx
Responsibilities: Provide an collapsable chat interface	

Server.js	
Parent class: Subclass:	Collaborator: SnakeUserData.route.js ChatRoute.jsx UserRoute.js db.js
Responsibilities: Set up the web server for chatbot	

ChatController.jsx	
Parent class: Subclass:	ChatController.jsx ChatModel.jsx ChatRoute.jsx Server.js
Responsibilities: Captures the chatbot logic generateResponse() createChat() appendMessage() getLatestMessage() getAllMessages()	

ChatRoute.jsx	
Parent class: Subclass:	ChatController.jsx ChatModel.jsx Server.js
Responsibilities: Handle POST requests for different API endpoints	

LoginPage.jsx	
Parent class: App.jsx Subclass:	Dashboard.jsx
Responsibilities: Allows users to sign up, change account and login HandleUserLogin() SwitchLogin(Signup()) HandleRegisterNewUser()	

Snake.jsx	
Parent class: Subclass:	SnakeGameState.jsx GameLoader.jsx User and snake scores databases
Responsibilities: Loading essential component providing user states Wrap the snake	

SnakeGameState.jsx	
Parent class: Snake.jsx Subclass: SnakeGameWindow.jsx	SnakeGameWindow.jsx
Responsibilities: Handling state change functions updating according to database	

SnakeGameWindow.jsx	
Parent class: App.jsx Subclass:	Snake.jsx
Responsibilities: Control the snake game logic	

User database	
Parent class: Subclass:	Snake.jsx Dashboard.jsx LoginPage.jsx
Responsibilities: Record the username, email, password, profile images Hashing passwords to protect users	

Folder Structure for member use only



User

snake

client

src

Components

Navbar
Footer
GameBoard
GameList
GameLoader
TipsLoader
PreviousAttemptsLoader



ChatBot



WordGame



SnakeGame



...Other Games

Pages

Index.html
App.jsx
HomePage
GamePage
LoginPage
Dashboard



Styles



Routers



Public

server

Controlers

UserController
ChatBotController
SnakeGameDataController
... OtherGameDataControllers...



config

Models

UserModel
ChatBotModel
SnakeGameModel
....OtherGameModels



middleware

Routes

UserRoute
ChatBotRoute
SnakeGameRoute
...OtherGameRoutes



utils



server.js

hashPassword

DB structure for member useonly



MongoDB



User



Chat



Snake



... for other Games