Hidden markov model

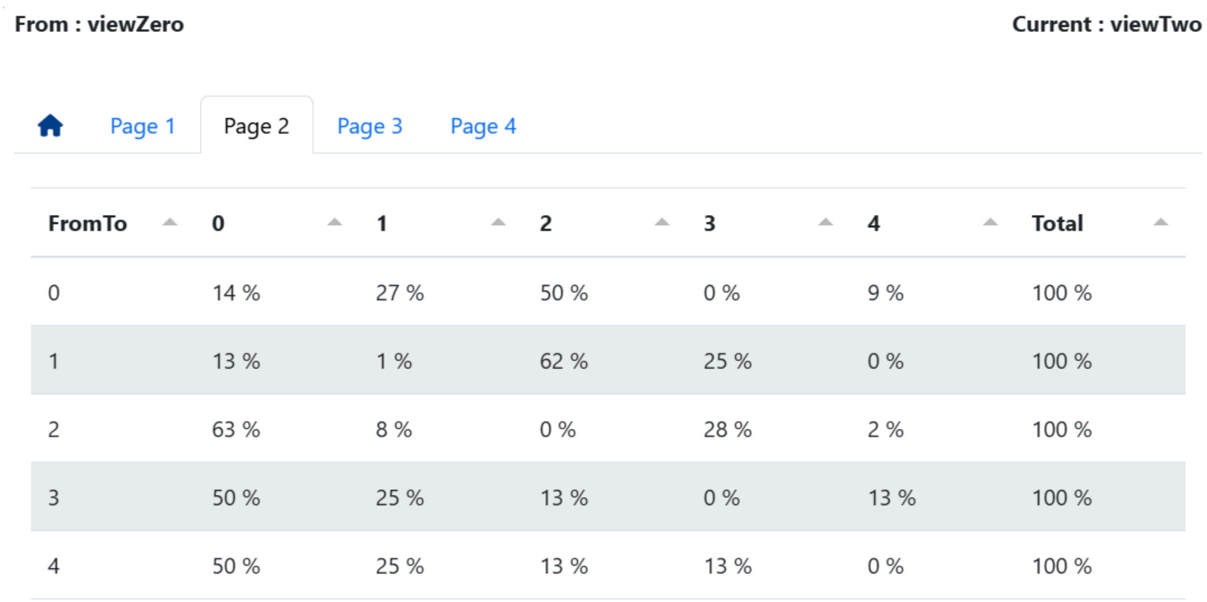
How HMM Works: a HMM goes by the idea that there is data in the underside of an application in which we can infer what the user do from where to where :

1. **States**: The possible hidden conditions or statuses (the different pages of a website).

2. **Observations**: The visible outputs that provide evidence of the system's state (clicks on a website).

3. **State Transitions**: Going from one state to another (going to a different webpage)

4. **Emission Probabilities**: The table of probably to get to a specific output (from a page to another)



From : viewZero

Current : viewTwo

FromTo	0	1	2	3	4	Total
0	14 %	27 %	50 %	0 %	9 %	100 %
1	13 %	1 %	62 %	25 %	0 %	100 %
2	63 %	8 %	0 %	28 %	2 %	100 %
3	50 %	25 %	13 %	0 %	13 %	100 %
4	50 %	25 %	13 %	13 %	0 %	100 %

In the context of tracking user on a website:

- **States** represent the different pages or sections of the website that a user might visit.

- **Observations** correspond to actions like clicks, which provide data about the user's action.

- **State transitions** get the percentage of probability that the user go from a specific page to another.

By studying the pattern of the state transitions, HMM can shows patterns, like frequently visited pages or chains of pages opened.

- **Predicting the next page**: HMM can be use to predict which page a user is most likely to go to from another page

- **Personalizing the user experience**: The website can be adapted to offer more logic to the users usage of the websites

This visualization of the table provides a clear view of user behavior and can be used to refine the website design by:

- **Highlighting mostly used paths** : Understanding which pages are most often visited together.

- **Adapting interfaces dynamically**: The site can change its layout or content based on user behavior patterns.

Advantages of HMM in User Behavior Analysis

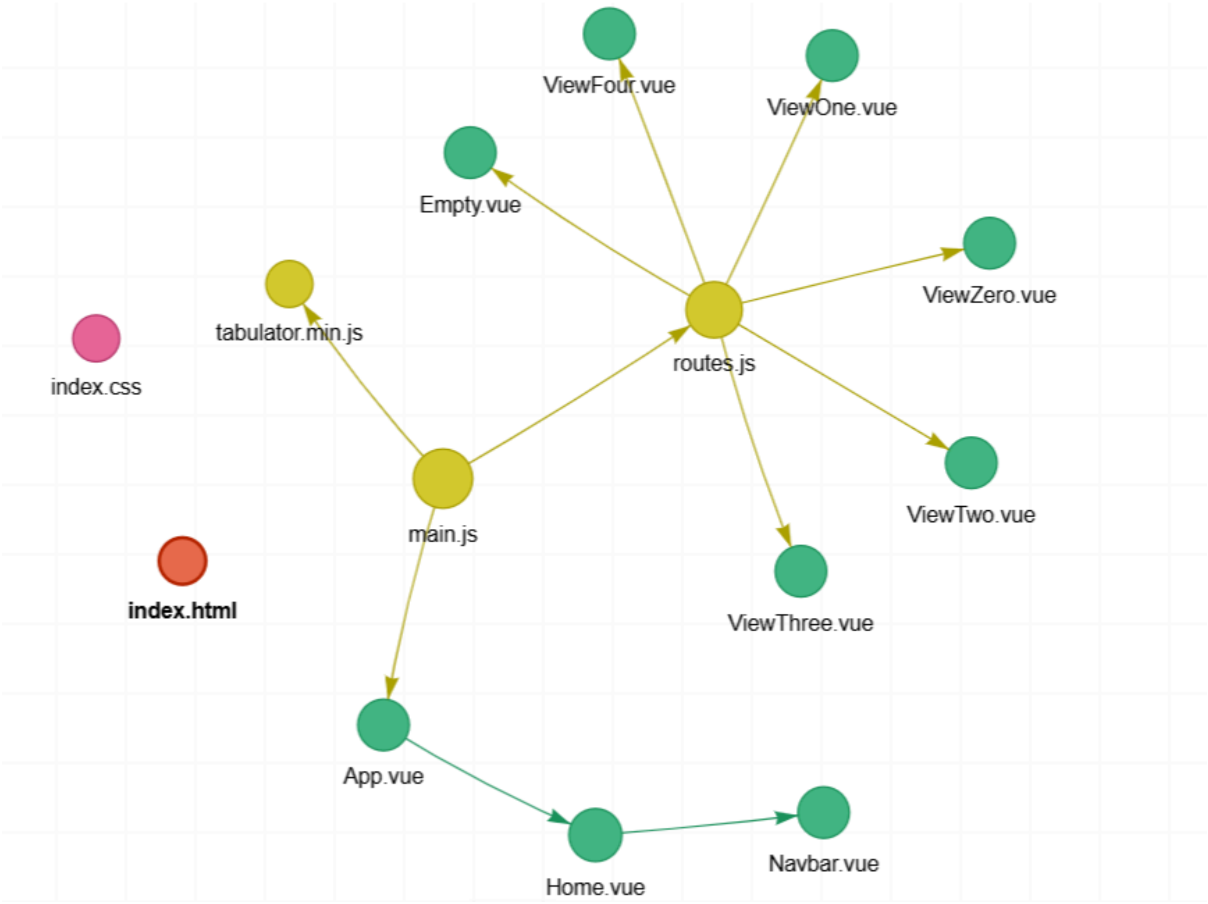
✔ **User Behavior Prediction**: Can anticipate the next actions of users with a certain probability.
✔ **Personalization**: Customizes the user experience by suggesting pages or actions based on prior interactions.
✔ **Data-Driven Decisions**: Provides insights into how users navigate the website, aiding in better design and decision-making.

By applying the Hidden Markov Model, we can continuously refine and improve the website interface, ensuring a smoother, more intuitive experience for users.

Explanation of the Technologies Used

Our application was developed using Vue.js, a progressive JavaScript framework that enables the creation of interactive and modular user interfaces. Vue.js facilitates component management and communication, allowing us to structure our application in a clear and efficient way.

In addition, Vue Router plays a crucial role in managing navigation between different views within the application. It allows us to handle dynamic routing.



<u>**App diagram Explanation**</u>

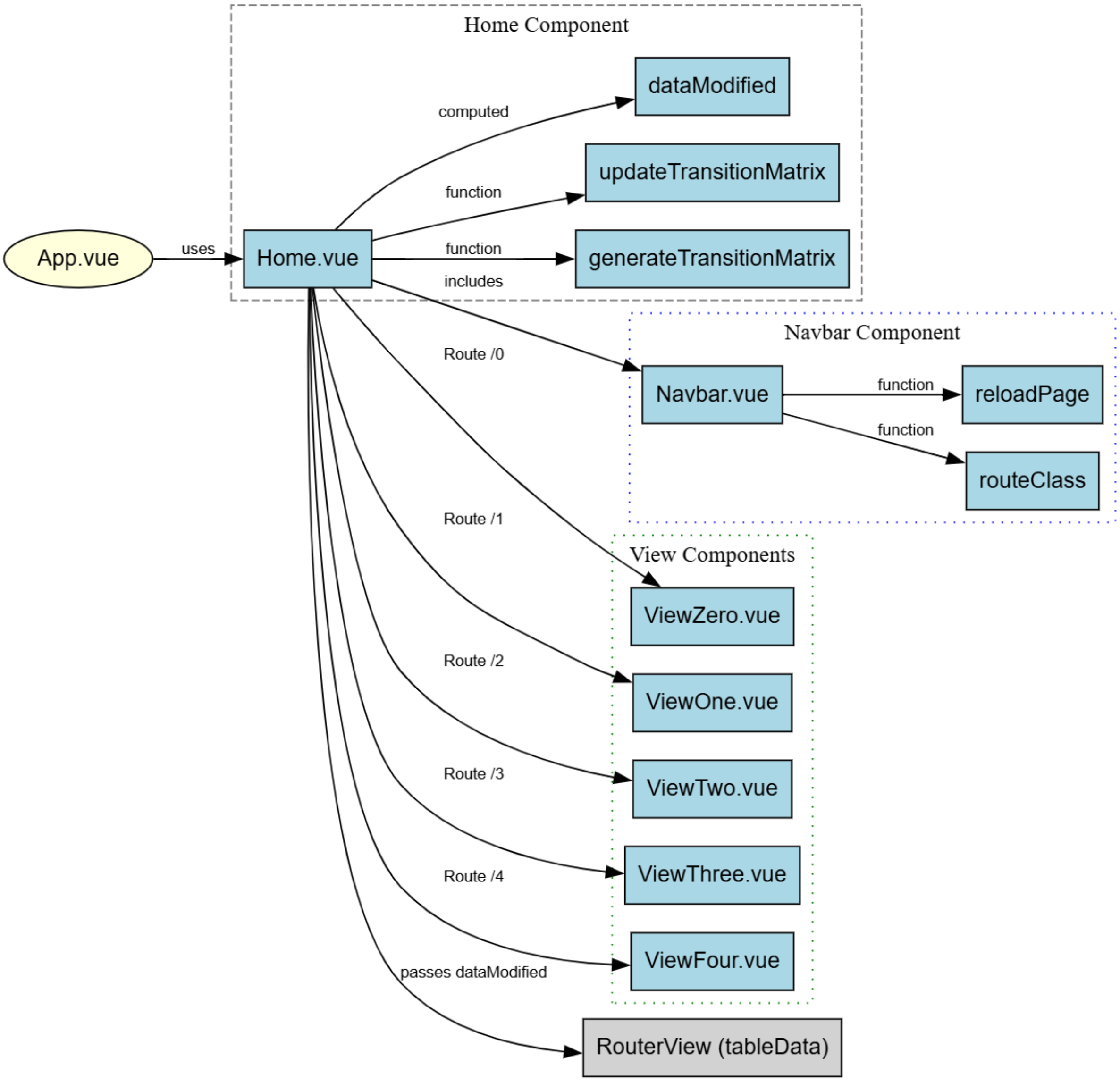


Diagram of the web app

This diagram illustrates the relationships and interactions between the main components of the application, as well as key functions and computed properties.

Main Components:

1. **App.vue**: The root component, importing and using the `Home` component.

2. **Home.vue**: Central to the application, it:

   ○ Displays a dynamic table via `RouterView`.

    - Manages a transition matrix using `generateTransitionMatrix` and `updateTransitionMatrix`.
    - Passes processed data to `RouterView` via the `dataModified` computed property.

3. **Navbar.vue**: The navigation bar component, providing navigation between views and offering functions like `routeClass` and `reloadPage`.
4. **View Components** (`ViewZero.vue`, `ViewOne.vue`, `ViewTwo.vue`, `ViewThree.vue`, `ViewFour.vue`): Display tables with data passed from `Home.vue`.

Functions & Computed Properties:

- **`generateTransitionMatrix`**: Generates a random transition matrix representing state-to-state probabilities.
- **`updateTransitionMatrix`**: Updates the matrix based on route navigation.
- **`dataModified`**: A computed property that formats the matrix data into a human-readable percentage format.