# **Assessment 02: React Native Hacker News App Assessment Rubric**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **10-9** | **8-7** | **6-5** | **4-0** |
| **Functionality & Robustness** | Application thoroughly demonstrates functionality & robustness on the following:   * Application opens without file structure modification. * Application runs without code modification. * Asynchronously fetch data from Hacker News using Axios & the Hacker News API. * Display the story’s title & score from the response contents as an item in a flat list widget. * Display the story’s URL from the response contents in a web view widget or redirects to a webpage. * Bottom navigation widget which navigates the user to the appropriate collection of stories. * Splash screen with an image view widget & transition animation. * Adaptive launcher icon. * Visually attractive user-interface with a coherent graphical theme & style. * Application is published to Google Play Store. * Ability to download the application from Google Play Store on to multiple mobile devices. | Application mostly demonstrates functionality & robustness on the following:   * Application opens without file structure modification. * Application runs without code modification. * Asynchronously fetch data from Hacker News using Axios & the Hacker News API. * Display its title & score from the response contents as an item in a flat list widget. * Display its URL from the response contents in a web view widget or redirects to a webpage. * Bottom navigation widget which navigates the user to the appropriate collection of stories. * Splash screen with an image view widget & transition animation. * Adaptive launcher icon. * Visually attractive user-interface with a coherent graphical theme & style. * Application is published to Google Play Store. * Ability to download the application from Google Play Store on to multiple mobile devices. | Application demonstrates some functionality & robustness on the following:   * Application opens without file structure modification. * Application runs without code modification. * Asynchronously fetch data from Hacker News using Axios & the Hacker News API. * Display its title & score from the response contents as an item in a flat list widget. * Display its URL from the response contents in a web view widget or redirects to a webpage. * Bottom navigation widget which navigates the user to the appropriate collection of stories. * Splash screen with an image view widget & transition animation. * Adaptive launcher icon. * Visually attractive user-interface with a coherent graphical theme & style. * Application is published to Google Play Store. * Ability to download the application from Google Play Store on to multiple mobile devices. | Application does not or does not fully demonstrate functionality & robustness on the following:   * Application opens without file structure modification. * Application runs without code modification. * Asynchronously fetch data from Hacker News using Axios & the Hacker News API. * Display its title & score from the response contents as an item in a flat list widget. * Display its URL from the response contents in a web view widget or redirects to a webpage. * Bottom navigation widget which navigates the user to the appropriate collection of stories. * Splash screen with an image view widget & transition animation. * Adaptive launcher icon. * Visually attractive user-interface with a coherent graphical theme & style. * Application is published to Google Play Store. * Ability to download the application from Google Play Store on to multiple mobile devices. |
| **Code Elegance** | Application code thoroughly demonstrates code elegance on the following:   * Idiomatic use of control flow, data structures & other in-built functions. * Sufficient modularity, i.e., UI split into independent, reusable pieces. * Handling of API response codes. * Header comments explain each function. * In-line comments explain complex logic. * Components adhere to composition & not inheritance. * Concise naming of components, functions & variables. * Efficient algorithmic approach. * Code formatted using Prettier or ESLint. | Application code mostly demonstrates code elegance on the following:   * Idiomatic use of control flow, data structures & other in-built functions. * Sufficient modularity, i.e., UI split into independent, reusable pieces. * Handling of API response codes. * Header comments explain each function. * In-line comments explain complex logic. * Components adhere to composition & not inheritance. * Concise naming of components, functions & variables. * Efficient algorithmic approach. * Code formatted using Prettier or ESLint. | Application code demonstrates some code elegance on the following:   * Idiomatic use of control flow, data structures & other in-built functions. * Sufficient modularity, i.e., UI split into independent, reusable pieces. * Handling of API response codes. * Header comments explain each function. * In-line comments explain complex logic. * Components adhere to composition & not inheritance. * Concise naming of components, functions & variables. * Efficient algorithmic approach. * Code formatted using Prettier or ESLint. | Application code does not does not fully demonstrate code elegance on the following:   * Idiomatic use of control flow, data structures & other in-built functions. * Sufficient modularity, i.e., UI split into independent, reusable pieces. * Handling of API response codes. * Header comments explain each function. * In-line comments explain complex logic. * Components adhere to composition & not inheritance. * Concise naming of components, functions & variables. * Efficient algorithmic approach. * Code formatted using Prettier or ESLint. |
| **Documentation & Git Usage** | README thoroughly describes how to set the environment for development.  Privacy policy thoroughly discloses user information collected by the application.  Application wireframes thoroughly sketched & reflect the final application.  Step-by-step user guide thoroughly describes each screen in detail.  Application code thoroughly commented with JSDoc & generated to Markdown.  Git branches thoroughly named with convention & contain the correct code relating to the functional requirement.  Git commit messages thoroughly reflect the functional requirement changes. | README mostly describes how to set the environment for development.  Privacy policy mostly discloses user information collected by the application.  Application wireframes mostly sketched & reflect the final application.  Step-by-step user guide mostly describes each screen in detail.  Application code mostly commented with JSDoc & generated to Markdown.  Git branches mostly named with convention & contain the correct code relating to the functional requirement.  Git commit messages mostly reflect the functional requirement changes. | README briefly describes how to set the environment for development.  Privacy policy briefly discloses user information collected by the application.  Application wireframes briefly sketched & reflect the final application.  Step-by-step user guide briefly describes each screen in detail.  Some application code commented with JSDoc & generated to Markdown.  Some git branches named with convention & contain the correct code relating to the functional requirement.  Some git commit messages reflect the functional requirement changes. | README does not or does not fully describe how to set the environment for development.  Privacy policy does not or does not fully disclose user information collected by the application.  Application wireframes are not or are not fully sketched & do not or do not fully reflect the final application.  Step-by-step guide does not or does not fully describe each screen in detail.  Application code is not or not fully commented with JSDoc & generated to Markdown.  Git branches are not or are not fully named with convention & do not or do not fully contain the correct code relating to the functional requirement.  Git commit messages do not or do not fully reflect the functional requirement changes. |

# **Assessment 02: React Native Hacker News App Marking Cover Sheet**

Name:

Date:

Learner ID:

Assessor’s Name:

Assessor’s Signature:

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Out Of** | **Weighting** | **Final Result** |
| Functionality & Robustness | 10 | 40 |  |
| Code Elegance | 10 | 50 |  |
| Documentation & Git Usage | 10 | 10 |  |
| **Final Result** | | | /100 |
| **This assessment is worth 25% of the final mark for the Design and Development of Application Mobile Devices course.** | | | |

Feedback: