What functionality you would build into this template?

- Settings: Most web applications consist of an interface, an application layer allowing users interact with data and a database. The first thing my application should include is a settings module/mechanism that defines how these parts are connected one to another. This includes database configurations, environment settings, server settings to override, etc.
- Authentication: a mechanism of securely authenticating users into the system should be set. Operations related to this also should be taken into account like password reset and recovery, account creation, etc. The right and most secure password hashing algorithms should be used, and user session management supported.
- MVC support: having worked with quite a few MVC frameworks I would implement it in my template in such a way that tables and entities from my database can be converted into classes that inherit from a base Model class acting as a container. A Controller class that handles operations that can be performed on models and view templates that make up the interface the user interacts with.
- API development: the template should allow developers to build an API that can easily be used by other applications to access and manipulate data given the user has the right permissions
- Data validation: user interactions with data will be monitored and validated before written to the database as one should never trust user input.
- Error handling and Logging: Handling errors will allow in the development phase debugging. Once in production, managing users and logging their activities will make troubleshooting and further analysis easier.
- Code generation: the template should come with a tool (CLI program or script) to generate controllers, models and view files, database migration files, etc.
- Admin panel: easy root data manipulation with little restrictions is needed in most cases. This will allow the administrators of the system to change data in instances where the end users are restricted to performs such operations.

What third party software would you include and how?

- Plugins and widgets: Since there are a lot of other functionalities that developers might need in their projects, I should allow other developers to develop and use plugins and widgets in the project.
- For better UI/UX it should be possible for developers to add their own JavaScript to the default views or totally replace them with their own using JavaScript, HTML/CSS libraries of their choice since the view files will be separate from the model containers and the functional controllers. By default, I would use Twitter Bootstrap, jQuery/Ajax, HTML5/CSS3 to build my template default widgets and views as they are light and have little to no dependencies or installs required.
- To support as many database systems as possible, I would choose how to connect my application to them depending on the language I will use to build the template. For the PHP language for example, SQL databases like MySQL, PostgreSQL will be supported by PDO and MongoDB by MongoClient which are built in the language. For any other DBMS not supported by the language external libraries might be needed.