DISCLAIMER: The overall focus of the app is on the field descriptions and the warping function. In this fail case manual, there is a generalised theoretical approach to mainly two class of problems: one is a set of standard or commonly known shapes, and the other is a functional approach towards guessing the possible sets of Prandtl stress functions. The algebra has been explicitly worked out in the problems till the point of obtaining the stress and the strain field descriptions, simply because in the process of integration of the strain descriptions to get the displacements produce a lot of unnecessary constants in the overall generalised expression. Since we have taken generalised expressions for almost all the cases and have also kept all other parameters (more or less) the same to generalise the approach and the process, the algebra has been worked out till the field descriptions only. However, since the app code has been designed once and for all, the app has the ability to calculate the displacement fields for any arbitrary case and also display surface plots on the app interface.

Following are a set of points one needs to keep in mind for getting a deeper and a finer sense of the topic and an insight into the working of the problems:

1. The first and foremost assumption (a note) that one must consider is: The shaft member is made of an isotropic material that is in the linear elastic domain (range). Hence, the behaviour of the material depends only on two properties viz. Young's Modulus E and Poisson's Ratio ν. These might not be obvious while giving the inputs since one uses the Modulus of Rigidity G for the class of problems under torsion in arbitrary shafts. However, the relation $G = E/2(1+\nu)$ shows that both E and ν are indeed involved in the calculations. That also means that the topics of failure criteria, yielding and plastic torsion are not to be discussed in this context.
2. Another important nuance about the functioning of the formulation of the problems is that St. Venant's principle is taken into account. That is, the cross-section that we're analysing is not a cut-section near the ends; it is a cut-section far enough from the ends to make the loading uniform and to ignore the end effects.
3. The app will not be able to distinguish between a function that represents a closed boundary i.e. a closed cross-section of the shaft and any other function (be it of polynomial form, sinusoidal or exponential!). This means, as far as the Laplacian is satisfied, the app will continue to assume that it represents an actual closed boundary and will produce results corresponding to that form of the function. Thus, there will be no physical meaning of the outputs shown in such a condition. However, for an invalid function (that does not satisfy the Laplacian), the app does give out a message saying "wrong input" in a check box.
4. The app takes the inputs and gives the outputs in SI units (e.g. the modulus of rigidity is to be filled in Giga-Pascals and the displacements calculated are in meters). However, in many questions there might be other units (which are more common in real life situations) that are used. Hence the app doesn't have the flexibility of unit conversion while solving the problem. The user will then have to manually convert the results (or the inputs) into SI units accordingly.
5. Since solving for a single arbitrary shaft is a complex problem in itself, the app has limitations regarding composite shafts with different arbitrary cross-sections (e.g. two slender shafts of different cross-sections joined to form a composite shaft).

6. The app is also made for solving problems on shafts of arbitrary cross-sections made of a single material and no variation in terms of material composition is accounted for, since that would make the problem extremely difficult to solve (even manually!)
7. For the fail case of a rectangular shaft, the Laplacian can be solved by breaking the equation into a particular and a general solution and the method ends up with a series solution to the Poisson equation. This is not shown explicitly as it is beyond the scope of the fundamental cases in torsion in arbitrary shafts. Instead, one can take narrow rectangular cross-sections as one of the cases that is seen in many of the engineering applications which is a passed case and hence can be exploited.
8. Problems on hollow elliptical (or circular) shafts can not entirely be solved by the app in one go. For a hollow member, the outputs will be given for the solid and the hollow sections, but the computation of the entire shaft needs to be done by the user manually (by subtracting the values associated with the hollow section from those associated with the solid section) to get the final values.
9. The app has a limitation on the calculation of torque on the shaft from the Prandtl function. Since the torque is related to the Prandtl function by the relation $T = 2\int \phi_p dA = 2\int \phi_p dx dy$ it has to be integrated twice and that becomes difficult when the function becomes implicit. Implicit functions will have a variation of their integration limits dependent on the coordinates. An examples would be $x : 0 \rightarrow 1$, $y : 0 \rightarrow 1 - x$
   In such scenarios, the integration limits need to be worked out for the specific problem at hand. That would require the app to recognise every possible scenario and the generalisation of the problem will become nearly an impossible task. Therefore, in general, for the arbitrary cross-sections, the torque is not considered in the parameters that are to be found.
10. For the equilateral triangle cross-section, although it is a textbook case, where even the torsional rigidity is also known in terms of the material properties and the dimensions of the cross-section, the app does not include it under the “Standard Shapes” tab since the Prandtl function will depend on the choice of the axes that the user defines i.e. the orientation of the triangle w.r.t the user-defined axes. The origin in the theoretical case has been taken at the centroid of the triangle. However it, may also be taken as one of the vertices or at any point along the edges of the triangle by the user and thus becomes very specific and adds complexity to the problem (such as the angle between the user’s axis and the symmetry axis of the triangle and the origin in both the reference frames). Therefore, the function corresponding to the axes chosen should be put in as an arbitrary function in the app.
11. One of the key assumptions while solving engineering problems and also from a design perspective is the assumption of a slender member - where the length of the member is *at least* 5 times greater than the other two cross-sectional dimensions (e.g. $\frac{L}{D} > 5$ where $D = 2R$ for a circle of radius R and length L and $\frac{L}{max\{2a,2b\}}$ for an ellipse of semi-major axis a, semi-minor axes b and length L). Since the length of the problem does not get involved into the process of calculating the field descriptions, the verification of the shaft member being a slender member is not possible with the given input options in the app. As a result, the app may give results for those cross-sections with dimensions that are comparable in magnitude with the length of the shaft member.